PB-211 068 130

THE COMBINED FILE SEARCH SYSTEM

developed for

THE FOOD AND DRUG ADMINISTRATION

by

THE SERVICE BUREAU CORPORATION

USER'S MANUAL

1 le

TABLE OF CONTENTS

INTRODUCTION 2

FILE DEFINITIONS 6

DICTIONARY SYSTEM 9

DICTIONARY LANGUAGE 19
DICTIONARY MASTER RECORD 22
TABBLD PROGRAM 23
DEDIT PROGRAM 28
DICT2 PROGRAM 34
DICTMSG PROGRAM 36
DPRINT3 PROGRAM 39
SYNTAPE PROGRAM 43
DPRINT4 PROGRAM 44
RELTERM PROGRAM 50
RELTRM1 PROGRAM 52

MAIN FILE SYSTEM 54

FILE MAINTENANCE LANGUAGE 64
MASTER FILE ORGANIZATION 72
MASTER FILE RECORD 77
INVERTED FILE RECORD 79
CROSS REFERENCE FILE RECORD 80
MFEDIT PROGRAM 81
MFMAINT PROGRAM 83
MPRINT1 PROGRAM 86
IDUPDATE PROGRAM 89
UPDXREF PROGRAM 91
XREFPRNT PROGRAM 93
IVUPDT PROGRAM 95
IFPRNT PROGRAM 97
IVPRINT PROGRAM 99
DUMPCELL PROGRAM 102
LOADCELL PROGRAM 104

SEARCH SYSTEM 105

SEARCH CONTROL PARAMETERS 108
SEARCH LANGUAGE 114
SEARCH PROGRAM 125

INTRODUCTION

The information storage and retrieval system developed by The Food and Drug Administration primarily for storing and retrieving information related to adverse drug reactions is also applicable to many other disciplines. Virtually any document whose contents can be accurately described by words or symbols (descriptors or index terms) and numeric data can be stored and retrieved by the system. In addition to storing the descriptive data contained in the searchable portion of a document, narrative text may be stored and recalled when that document satisfies the criteria specified in a search request.

The system is designed for manual indexing where an indexer describes the significant portions of the contents of a document through the use of Descriptors or Keywords. To refine the meaning of the descriptor, we have introduced the qualifier. A qualifier may consist of a subdescriptor, a numeric value, or a combination of both.

Since the system is designed to operate in a direct access environment, care must be taken not to bulk the files unnecessarily. For this reason, several types of descriptors are defined in the system. They are: PRECISE, COMMON, TEMPORARY, and ID.

PRECISE:

A Precise descriptor is a keyword which generally doesn't occur is a high percentage of the documents in the system and hence distinguishes a document as different from most other documents. A useful way to determine if a descriptor should be designated as precise is to consider how many documents would be retrieved if a query was to be made based only upon that particular descriptor. If the number would be very excessive, the term should probably, although not necessarily, be Common.

COMMON:

In contrast to the precise descriptor, a Common descriptor is one that is contained in too many documents to be very discriminating for searching purposes. For example, asking for all MALES in a medical data base would result in retrieving about one-half of the file. That very question can be handled by the system, but not by the classical inverted file approach.

TEMPORARY:

When new terminology is added to the system's vocabulary, it may not be predetermined whether a term should be designated as precise or common since not enough is known about the frequency with which the term will be used as a descriptor. In this case, it would be assigned a temporary precise function and could be reclassified later.

ID.:

All descriptors validly indexed into documents resident in the system are contained in the searchable portion of the document file records. In addition, common descriptors are extracted for inclusion into the identification (ID.) file records. All records in the document file containing any common descriptors will produce a corresponding entry in the ID. file. The ID. file is searched when a query is made based solely upon common descriptors. In addition to the common descriptors, the user can specify other descriptors to be included in the ID. file by classifying them as ID. file descriptors.

A number of keyword oriented systems fail to carry the intelligence related to the context in which a descriptor has been used. For instance - if a record contains the descriptor RASH, how does the system tell whether RASH is a symptom for which a drug was administered or a reaction caused by a drug. This system makes such a distinction possible through the introduction of file/context codes which may be associated with a descriptor either by the indexer or automatically by the dictionary.

In a well implemented information storage and retrieval system, care must be exercised to assure proper use of the indexer's vocabulary. Once an improperly indexed document enters the system, it may be extremely hard to find. Obviously, the system doesn't detect the improper use of a term from a conceptual point of view, but it does diagnose misspelling, use of a term as a descriptor when it is not allowable as such, indexing a document with a descriptor which is not valid for that particular data file, and use of a non-preferred term (in which case the preferred term is automatically substituted).

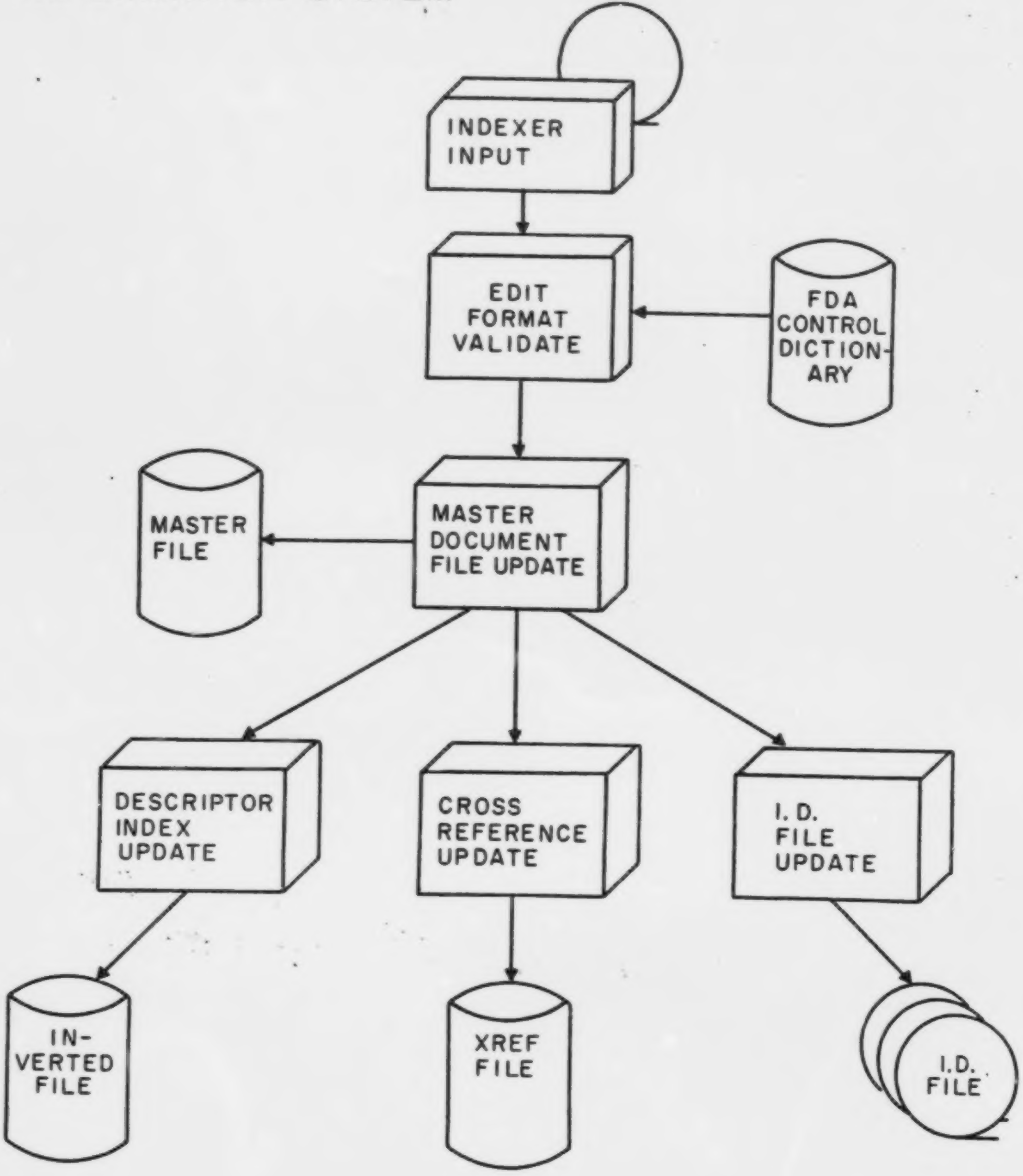
Vocabulary control is accomplished through the use of a system dictionary. The dictionary contains the vocabulary of index terms (descriptors) which the indexer may use in describing documents. Each index term is entered in the dictionary in ascending alphabetic sequence. Subordinate to each term are certain related data such as:

1. List of dictionaries to which the term belongs

2. List of file/context codes
3. Preferred or USE term (if a synomym exists)
4. Related terms (if any occur in the dictionary)
5. See also terms (as desired)
6. Scope notes (as desired)

Related terms are treated in a special way in this system. Any related terms stored under a main term are automatically added to a document whose contents are described by that main term. This amounts to a form of semi-automatic indexing.

INFORMATION SYSTEM



- ***MASTER FILE***

TERM ________
TERM ________ QUALIFIER ________
TERM ________

- ***INVERTED FILE***

TERM # # #
TERM # # #

- ***CROSS REFERENCE FILE***

EXT # INT #
EXT # INT #

- ***I.D. FILE***

COMMON ________
COMMON ________
COMMON ________

The System maintains five files which are basic to the system. They are:

1. Dictionary File

   The dictionary is primarily designed to control the vocabulary used by the indexer and to serve as a guide to the searcher in formulating requests. An important feature of the dictionary maintenance subsystem is the ability to produce a variety of listings which may be used by personnel maintaining the vocabulary and can be distributed to the users of the system. The listings may also be a useful aid in standardizing a vocabulary related to a particular discipline such as the EJC thesaurus developed by the Engineering Joint Council.

2. Document File

   The document file is the heart of the retrieval system since it contains the documents stored in the system. All descriptors used in indexing documents (whether they are precise, common, or temporary), their context codes, their subdescriptors and numeric values, and their associated links are stored in the searchable portion of the document file records. Textual information realted to a document is stored in "free text" segments which immediately follow the searchable segment.

3. Identification File (ID. File)

   The ID. file records are subsets of the information contained in the searchable segment records of the document file. All common descriptors and their qualifying data are contained in the ID. file for any given document along with other descriptors which have been specifically tagged for entry into the ID. file.

4. Inverted File

   The inverted file is organized in ascending alphabetic sequence by descriptor followed by a list of all documents in which the descriptor was used as an index term. If a descriptor was used in 500 documents, a list of 500 document numbers would be appended to the descriptor. Although common descriptors appear in the inverted file, a list of their document numbers would not be included. The inverted file serves as a high speed screening aid to be used by the search programs.

5. Cross Reference File

When a document is indexed into the system, it is assigned an "internal document number" which will differ from the manually assigned document number in almost all cases. Only coincidence would cause them to be identical. In order to allow the searcher to query based upon an 'external" or manually assigned document number correspondence must be maintained between the manually assigned document number and the number assigned by the file maintenance system. Also, duplicates being added to the system are rejected by means of determining whether the document already exists within the file. This option may be overridden by the indexer if duplication of external numbers is desirable.

The system allows several types of searches to be performed:

1. Document Search

In the document search, the searcher must enter a list of the document he wishes to retrieve. He may request the documents by either internal or external numbers. If the request is by external number, the first number in the list must be preceded by an 'X'.

2. Mixed Search

In the event the searcher may wish to retrieve a predetermined list of documents which satisfy some logical criteria, he enters the document numbers followed by a Boolean logic statement representing the logical constraints which must be satisfied by the contents of the documents.

3. Boolean Search

The Boolean search is used when the searcher wants to retrieve all documents in the system which satisfy a specified list of criteria. In this case, the searcher specifies the list of criteria necessary to be satisfied for a document to be retrieved in the form of a Boolean logic statement.

In designing a generalized information retrieval system, one cannot conceive of all of the reports a user may wish to derive from the system. For this reason the system has a variety of standard printouts which may be obtained. In addition to the standard printouts, the searcher may request a report tape which will contain selected data in a fixed format which he may process by use of his own specialized programs.

THE DICTIONARY SYSTEM

DICTIONARY SYSTEM

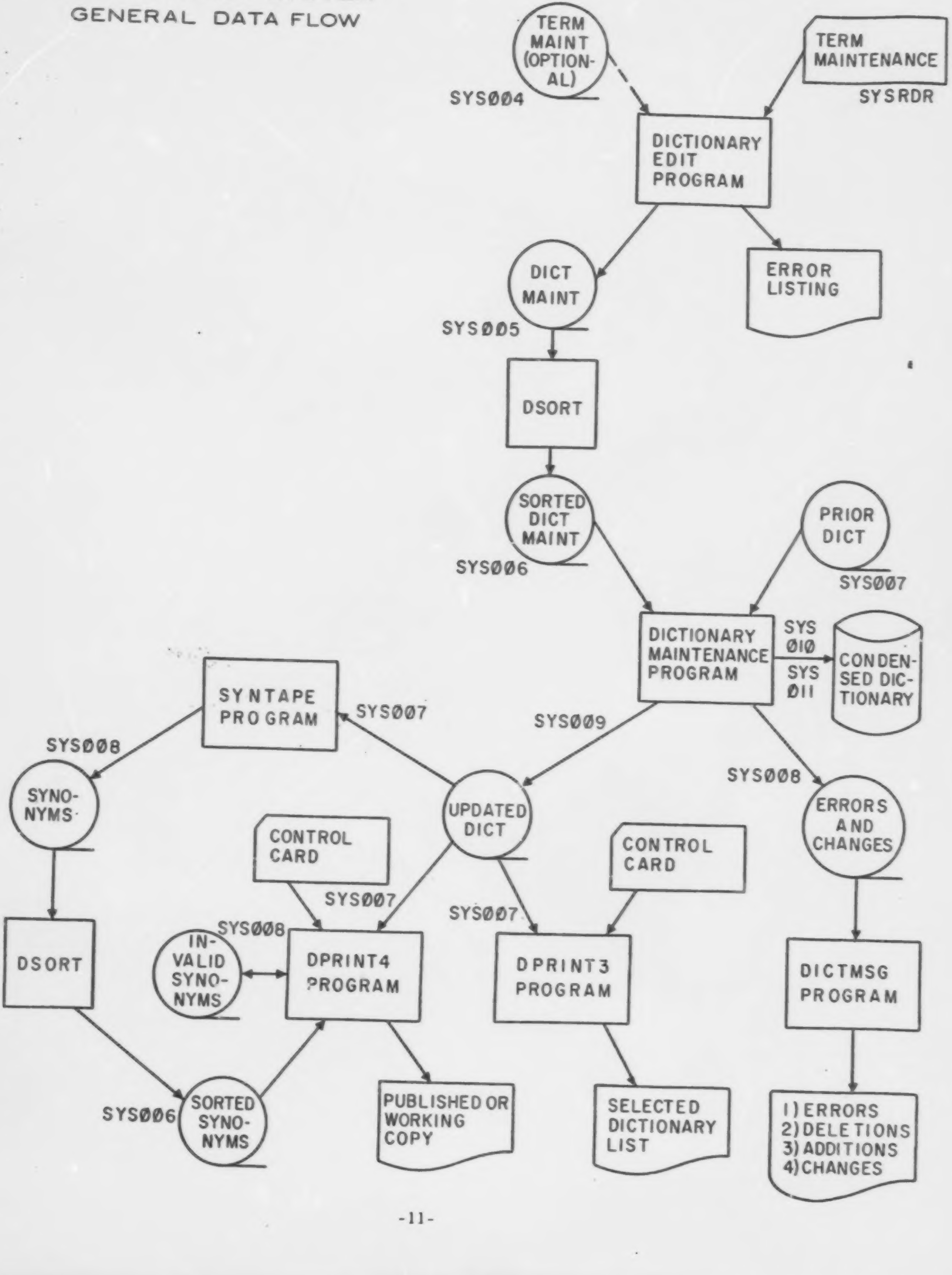
This series of programs, known collectively as the dictionary system, is designed to produce tape, printed, and disk file versions of the dictionary terms comprising the vocabulary used in the information storage and retrieval system.

The dictionary serves two major functions within the total system. First it is used as an automatic validation device against which both new documents entering the system and requests for information from the system are checked. This validation ensures that only those terms which have been approved as acceptable descriptive terms are used to describe the documents, and likewise, that inquiries to the file are phrased in acceptable terms. The second major function of the dictionary is to provide printed copies of the acceptable terms for use by document indexers, requestors, and other interested parties. Terms may be added to, deleted from, or changed within the dictionary and new versions prepared (in either or both machine readable and hard copy form) as desired.

The system consists of a series of nine programs which perform the various functions required. They are:

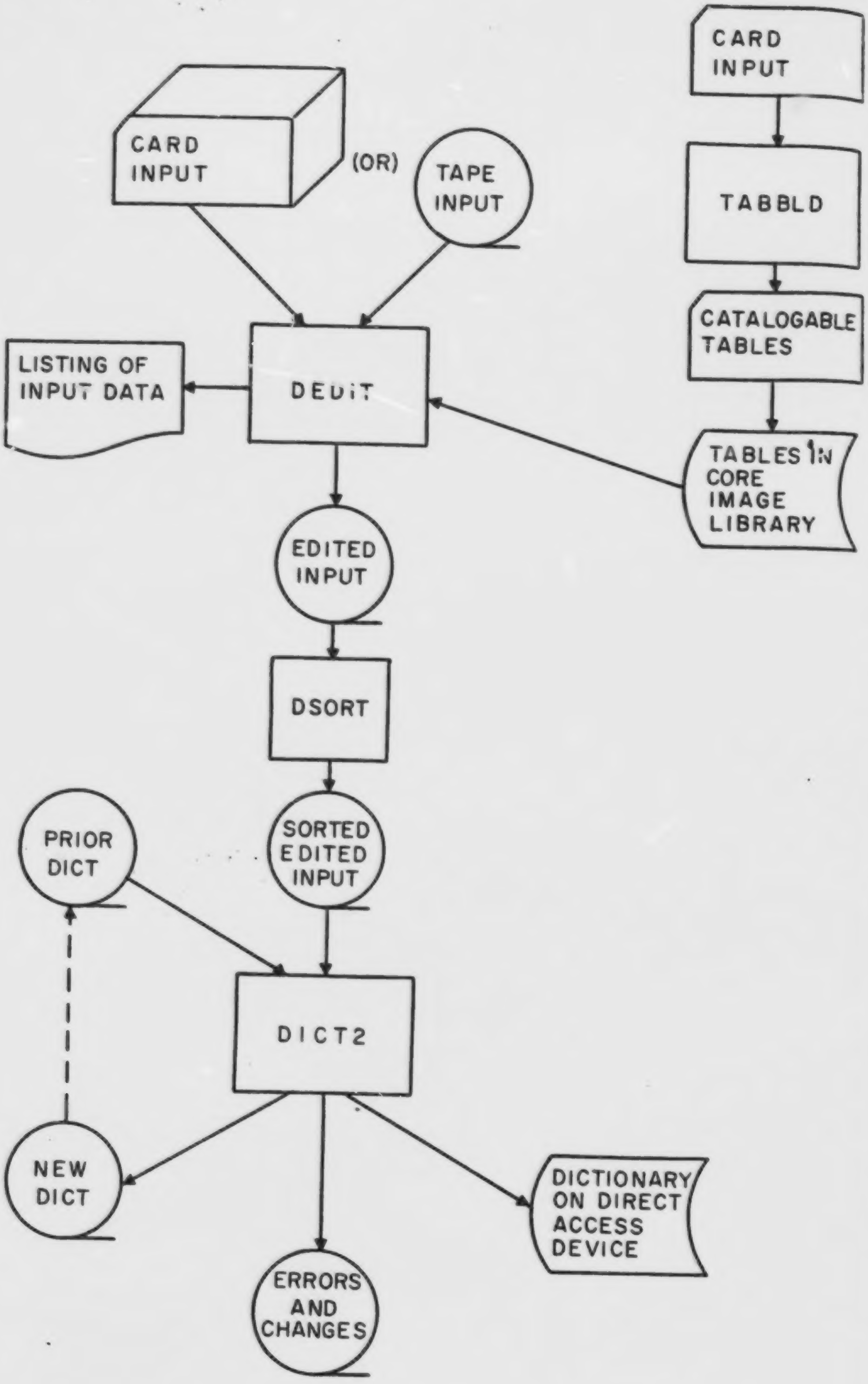
1.	TABBLD	(Table Builder)
2.	DEDIT	(Dictionary Edit)
3.	DICT2	(Dictionary Maintenance)
4.	DPRINT3	(Dictionary Print)
5.	DPRINT4	(Dictionary Print)
6.	SYNTAPE	(Synonym Tape)
7.	DICTMSG	(Dictionary Error Message Print)
8.	RELTERM	(Subordinate Term Extractor)
9.	RELTRM1	(Subordinate Term Validator)

DICTIONARY SYSTEM GENERAL DATA FLOW



The basic operations of the system are handled by two programs: DEDIT (Dictionary Edit) and DICT2 (Dictionary Update). DEDIT accepts the punched cards (or tape equivalent) representing the data to be placed in the dictionary. It validates the data, ensuring that various conditions are met, and converts it into a form suitable for further processing. Errors are printed as encountered in the data.

DICTIONARY MAINTENANCE

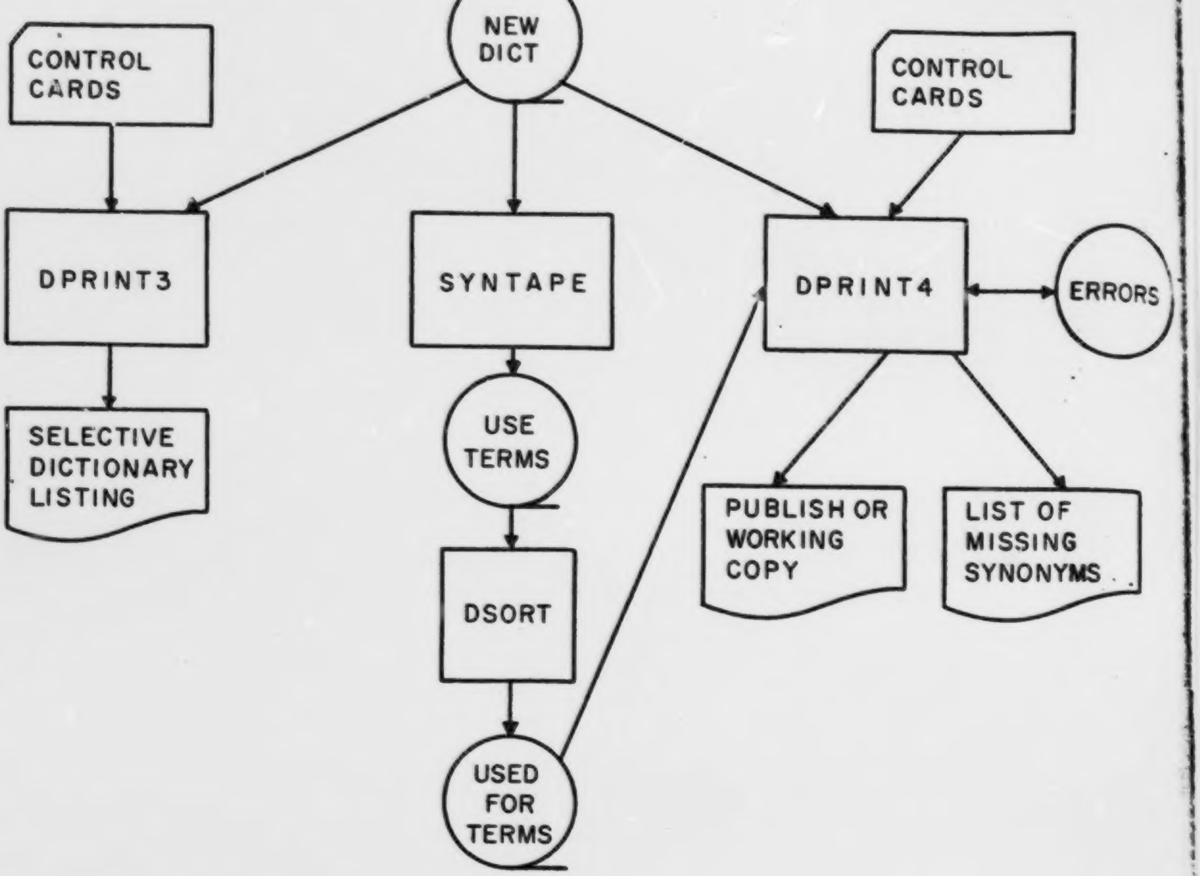


After the edited information has been sorted in sequence using a standard sort program, it is ready for processing by DICT2. This program accepts the edited input tape and the previous dictionary master tape (if any). The output from DICT2 consists of a new master dictionary tape, a new disk file version of the dictionary, and a tape of dictionary changes which resulted from the combining of the new input with the previous dictionary file. This change tape also contains error messages for invalid data, if any. A schematic illustration of the functions of DEDIT and DICT2 is shown below.

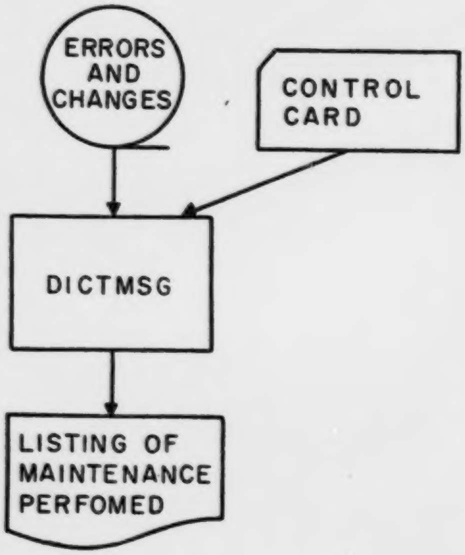
One of the functions of the DEDIT program is to convert the three letter mnemonic codes used to identify the various dictionaries within FDA (e.g., HUMan, VETerinary, etc.) into numbers, since these codes are stored as numbers within the files of the system. For this purpose, the DEDIT program requires access to a table of equivalents which gives the number corresponding to a given three letter code. The table can be created, modified, or added to at any time by use of the TABBLD (Table Building) program. This program accepts up to 256 data cards containing the alphabetic codes and their numeric equivalents, builds a table of this data, and stores it on the disk file where it is accessible to the DEDIT program. This table must be loaded into the system before DEDIT can be executed. TABBLD must be, therefore, the first program run when this dictionary system is initially established. It may, of course, be run again at any later time to add to the alphabetic-to-numeric code assignments if desired. Normally, however, it is not expected that TABBLD will be run with any great frequency. Usually, a single execution to establish the table intially, with only occasional executions later to add new dictionary codes to the table should be required.

These three programs, then, create the machine readable dictionary for FDA. TABBLD provides alphabetic-to-numeric equivalents for DEDIT which creates the dictionary records. DICT2 processes those dictionary records producing an updated dictionary tape, a disk file version of the dictionary, and a tape record of the changes which have been made. The requires six programs in this series provide a number of different ways in which this data can be printed, and a means of ensuring that internal consistency within the dictionary is maintained.

DICTIONARY PRINTING



MAINTENANCE LISTING



The program DICTMSG is used to print the tape of changes, deletions, additions, and errors resulting from execution of DICT2. Certain options are provided in DICTMSG to permit printing of only certain classes of data (e.g., print only the additions, or only the errors, if desired). DICTMSG is intended only as a maintenance tool, to provide printed indications of what changes have occurred. It is not intended to produce "publishable" copy.

DPRINT3 accepts the dictionary tape as input and produces a printed copy of the dictionary. This program, unlike DICTMSG, operates on the full dictionary tape, and can print all dictionary entries. It is provided with a number of options to permit selective printing of only certain entries, or of only selected portions of data from entries. It can, for example, print only the SEE ALSO data from selected entries.

DPRINT4 is the program which produces publishable copy from the master dictionary tape. It produces output in two formats: 8 1/2" x 11" pages and 14 x 22 inch pages. The latter is designed for photo-reduction and subsequent publication. DPRINT4 can be used in two different modes. In the first mode, the dictionary tape is simply printed as it appears on tape. In the second made, "reverse cross references" are included. For example, the dictionary may contain the following three entries:

```
ACETYLSALICYLIC ACID

HELICON      use ACETYLSALICYLIC ACID

MEASURIN     use ACETYLSALICYLIC ACID
```

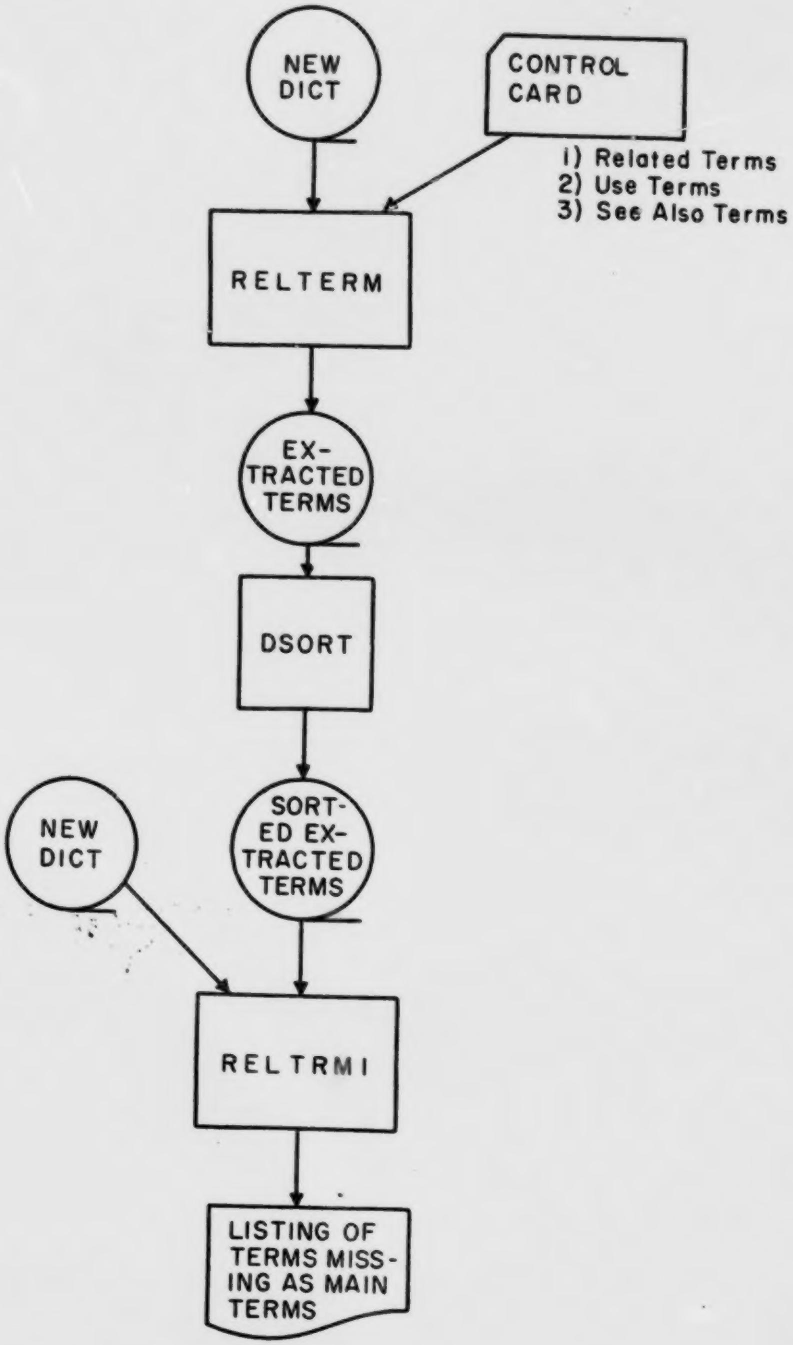
In the first mode, those three entries will appear as shown above. However, in the second mode, the ACETYLSALICYLIC ACID entry will appear as follows:

```
ACETYLSALICYLIC ACID

      used for    HELICON

      used for    MEASURIN
```

DICTIONARY VALIDATION



This has the effect of bringing together under the preferred term (Acetylsalicylic Acid in this case) all the other terms which refer to it. In order to locate and sequence these "see references" so that they will appear under their appropriate preferred term, antoher program, SYNTAPE (synonym tape generator) is used, followed by a standard sort. The sort produces a tape which is then merged with the master dictionary tape, producing a combined listing of both the dictionary entries themselves and their synonyms.

The subordinate term validation programs ensure that each term used in the dictionary as a related term, a use term, or a see also term occurs as a main term as well. This maintains integrity within the dictionary, eliminating blind references because of misspelling or revision of terms. RELTERM extracts the subordinate terms and RELTRM1 matches each with a main term. Any terms not found as main terms are listed with the main term under which they occur.

DICTIONARY LANGUAGE

The Dictionary Language is designed for the use of the editorial staff charged with the maintenance of the FDA Dictionary System. It offers a means of adding new terms to the dictionary, changing the various elements comprising an entry, and of deleting which are no longer required terms.

The dictionary consists of a variable number of "entry terms", each of which is followed by some descriptive information, known as the "term data". An illustrative sample of an entry term followed by term data is given below.

```
ABBOCILLIN
  FUNCTION-- PD,SD      DICTIONARIES--115
  FILE / CATEGORY NUMBERS--
    163-157,1-14

  RELATED TERMS--
    ANTIBIOTICS                    C PENICILLIN G POTASSIUM-C
    P PENICILLINS                  PROCAINE PENICILLIN
  SCOPE NOTES--
       MAJOR INDEX TERM  ABBOCILLIN
       AUTH-M
```

In this illustration, ABBOCILLIN is the entry term. The information following it is the term data. In this particular case, the following elements are present.

FUNCTION--PD,SD	Indicates that this term is valid for use as a precise descriptor or a sub-descriptor
DICTIONARIES--115	Indicates that this term is a member of the FDA Dictionary identified as number 115
FILE/CATEGORY NUMBERS-- 163-157,1-14	Indicates that this term has been assigned the File/Category (context) codes shown
RELATED TERMS-- ANTIBIOTICS--etc.	These terms have been assigned to ABBOCILLIN because they are closely related in their meanings and usage within FDA. Every time ABBOCILLIN is used to describe a document indexed into the master file, these related terms will automatically be assigned to that same document

SCOPE NOTES--	Variable contents, as assigned by the individual who entered this term in the file
MAJOR INDEX TERM AUTH-M	ABBOCILLIN

Another type of term data is permitted as well: SEE ALSO references. They are similar in form and content to RELATED TERMS, but are included only for the indexer's benefit in printed versions of the dictionary. They are not automatically included in the indexing of any given document. This particular term has no SEE ALSO references.

The dictionary editor may use any or all of the above elements to describe a dictionary term. The only one which is mandatory is FUNCTION. The editor must indicate whether a term is a precise, a common, or a temporary term so that the term will be handled in the proper manner when incorporated into the indexing of some document.

In addition to entries of the nature shown above, a second kind is permitted, the 'USE' reference. An example of such an entry is shown below.

```
NAPTHYL SALICYLATE
  FUNCTION-- PD           DICTIONARIES--ALL

  USE--

    NAPTHOL SALICYLATE
```

The entry for the term referred to is shown below.

```
NAPTHOL SALICYLATE
  FUNCTION-- PD           DICTIONARIES--115,2
  FILE / CATEGORY NUMBERS--
    1-66,163-179

  RELATED TERMS--
    ANTISEPTICS
    SALICYLATES
  USED FOR--
    NAPTHYL SALICYLATE
  SCOPE NOTES--
       MAJOR INDEX TERM- A-NAPTHYL SALICYLATE
       AUTH-M
```

If NAPTHYL SALICYLATE is ever used by an indexer to describe some document being added to the FDA file, the dictionary programs will automatically substitute NAPTHOL SALICYLATE.

The 'USED FOR' entry under NAPTHOL SALICYLATE is of particular interest. These USED FOR entries are never established by the dictionary editorial staff. They are an automatic result of the execution of certain of the dictionary programs (SYNTAPE and DPRINT4). Every 'USE' reference is automatically inserted in its appropriate place under the term referred to, so that all synonymous terms are brought together for inspection under the preferred term.

If an editor should inadvertently construct a 'USE' reference referring to a term which is not in the dictionary, the programs will detect this error and print a message to that effect. Every term referred to in a 'USE' entry must be an entry in the dictionary. It must, of course, not be another 'USE' entry.

In preparing input for the dictionary maintenance system, the editors will use a "free-form" card format, in which various elements of the information may be entered in a reasonably unconstrained manner. Careful attention must be paid to the punctuation, however, because these punctuation marks are used to distinguish the information itself from the "controldata" which indicates the kind of information being dealt with. The details of these input formats are contained in the description of the Dictionary Input Edit Program. The internal format of the dictionary record in the system is shown below.

DICTIONARY RECORD

LENGTH	CONTENTS
2 bytes	Record Length
2 bytes	Binary Zeros
20 bytes	Unused
20 bytes	Dictionary Codes
2 bytes	Disk Record Length
2 bytes	Binary Zeros
1 byte	Record Content Code
1 byte	Main Term Function
1 byte	Number of File/Context Pairs for Main Term
0-510 bytes	File/Context Codes
1 byte	Main Term Length +3
1 byte	Main Term Length +2
1 byte	Main Term Code
$1 \le N \le 64$ bytes	Main Term
1 byte	Use Term Length +3
1 byte	Use Term Length +2
1 byte	Use Term Code
$1 \le N \le 64$ bytes	Use Term -1 Only
1 byte	Related Term + File/Context + 3 Length
1 byte	Related Term Length +2
1 byte	Related Term Code
$1 \le N \le 64$ bytes	Related Term
0-510 bytes	Related Term File/Context Codes
1 byte	See Also Term Length +3
1 byte	See Also Term Length +2
1 byte	See Also Term Code
$1 \le N \le 64$ bytes	See Also Term
1 byte	Scope Note Length +4
1 byte	Scope Note Length +3
1 byte	Scope Note Code
1 byte	Scope Note Line Number
$1 \le N \le 60$ bytes	Scope Note

TABLE BUILDING PROGRAM (TABBLD)

Purpose

TABBLD is used to build tables of dictionary, file, and context mnemonics. After determining the validity of the input data, it sorts that data and then punches the complete job stack deck to permit loading and cataloging of the tables required for the dictionary input edit program (DEDIT).

Input

1. Requirements

   A control card must come first, followed by from 1 to 256 input cards. The dictionary table may have only 255 entries.

2. Format

   a. Control Card

      For each run there must be one and only one control card. There are three permissible formats, all of which begin in column 1. They are:

      1. DCT*
      2. FIL*
      3. CAT*NNN where NNN represents a file name. Please refer to the description of input cards for a more detailed specification.

   b. Input Cards

      Input cards have the following format:

      1. NNN XXX where NNN represents the file name mnemonic. The first character which appears in column 1 must be alphabetic. The other characters may be alphabetic or numeric with trailing blanks. No special characters or embedded blanks are allowed.

Column 4 is left blank.

XXX which is punched in columns 5 through 7 symbolizes any number from 000 to 255. Each number must be used only once in each data set, but the numbers need not be in sequence. For the dictionary table, the number may not be zero, since zero is used to indicate that a term is included in all dictionaries.

The remaining columns will normally be left blank, but they may be used for comments. The program does not check them in any way but they will appear on the printed listing of input.

Processing

The program reads the first card which must be the program control card and set up the 'phase' card in the output format area. It tests whether the card is a DCT*, or CAT* card and, if it is a CAT* card, checks that the filename mnemonic (columns 5 through 7) is valid. If no valid control card is found, the program prints an error message and cancels the job.

When a valid control card has been found, the program reads and prints the input cards. A running count is kept to check that no more than 256 cards are entered. As each card is read, it is checked to be certain that the number in columns 5 through 7 is between 000 and 255 and is unique and in the current data set. The filename mnemonic in columns 1 through 3 is checked for validity. Whenever an error is encountered, the program sets a switch and prints a diagnostic message. If no errors have been found in the set, the first seven card columns are stored. The program continues to read, print, and test the input cards until the end-of-file is reached.

After all the input cards have been read and processed, the program tests whether any errors were logged. If there were errors, a message is printed and the job is cancelled.

If no input errors were detected, the program performs an internal sort on columns 1 through 3 of the input. If identical mnemonics are found, a switch is set.

After the sorting has been completed, the program tests whether any mnemonics were repeated, and if so, determines which ones were duplicated, prints an error message, and cancels the job.

If there was no duplication, the program tests whether 256 cards were entered and, if not, arranges to 'pad' the punched output. The dictionary input edit program requires that the tables have 256 entries. 'Padding' is simply the filling of sufficient extra positions to bring the total to 256.

It then sets up and punches the complete deck required for loading and cataloging of the tables used in the dictionary input edit program.

Output

1. Card Formats and Explanatory Notes

   a. Card Formats

1.	// JOB CATALOG	
2.	// OPTION CATAL,LINK	
3.	PHASE DCTTAB,* PHASE FILTAB,* PHASE CATNNN,*	The characters in columns 10 through 12 correspond to columns 1 through 3 of the input control card. NNN represents the filename mnemonic characters which appear in columns 5 through 7 of a CAT* type input card.
4.	// EXEC ASSEMBLY	
5.	CSECT	
6.	DC C'XXX' DC FL1'XXX'	The XXX's in this pair of cards represent the filename mnemonic and corresponding number. The number of pairs of DC cards will be equal to the number of input cards.
7.	DC NNNX'FFFFFFFF'	A filler card is punched only if there were fewer than 256 input cards. NNN equals the difference between 256 and the number of input cards.
8.	/*	
9.	// EXEC LNKEDT	
10.	/&	

2. Listing of Sample Output

For the protection of the user, this program produces punched output only when all of the input was valid. A listing of sample output is given on the following page.

Listings of Sample Output

1. Valid Input

```
CAT*VET

PHB 002                THIS SPACE MAY BE USED FOR COMMENTS
PHA 001
CHC 004
ING 003
MIT 006        COMMENTS WILL NOT BE CHECKED IN ANY WAY BY THE PROGRAM
DRG 005
```

2. Invalid Input

```
PHA 001
 HB 002
     IT IS INVALID TO USE SPECIAL CHARACTERS AND LEADING OR EMBEDDED BLANKS IN COLUMNS 1-3.
      ONLY ALPHABETIC OR NUMERIC CHARACTERS AND TRAILING BLANKS ARE ALLOWED.

MIT 299
     THE NUMBER IN COLUMNS 5-7 MUST FALL BETWEEN 000 AND 255.

C02 004
DRG 001
     ANY GIVEN NUMBER IN COLUMNS 5-7 MAY BE USED ONLY ONCE IN EACH DATA SET.  THE NUMBER 001 HAS BEEN
     REPEATED.
AB/
     THE NUMBER IN COLUMNS 5-7 MUST FALL BETWEEN 000 AND 255.
```

DICTIONARY EDIT (DEDIT)

Purpose

The function of DEDIT is to examine the input information submitted on card or tape records. When the data and operations involved have been evaluated, the information is rewritten in a standard format. If all information for a term is acceptable, the records formed for it are written on tape. Otherwise, errors which have occurred are noted on the printed output.

DEDIT permits two broad classes of changes to be made to the dictionary file. Individual dictionary entries can be added, deleted, or modified as desired. In addition, words or phrases used as related terms, see also references, and the like can be revised or deleted wherever they appear through use of the 'block change' feature.

Input

Input to this program is punched into cards according to the formats described below. General constraints which apply to DEDIT input are:

1. A dictionary term may consist of not more than 64 alpha-numeric characters. Seven special characters are also permissible: right and left parentheses, hyphen, slash, comma, period, and pound sign.

2. All input cards must be in ascending sequence based upon digits punched in columns 1 through 4. These numbers need not be consecutive.

3. All block change cards must precede any changes to individual dictionary records. Block change cards may be numbered independently from individual change records (i.e., block change cards may be numbered from 1 to n, immediately followed by individual term changes numbered from 1 to n).

4. The first input record must be a BCH (block change header) record or a HDR (dictionary term header) record.

5. All ADD, DELETE, and REPLACE records may include continuation cards. These continuation cards contain a sequence number in columns 1 through 4, with columns 5 through 10 blank. The continuation data begins in column 11.

6. If there is a character punched in column 80 of a card and a character punched in column 11 of a continuation card, it is assumed that there is no blank between the two characters.

7. All SCOPEnn (scope note) cards must be the last cards pertaining to a given dictionary term. Only another SCOPE card, relating to the same term, a HDR card - indicating the beginning of a new dictionary term--or an EOF (end-of-file) card may follow a SCOPE card.

Block Change Cards

1. Block change header must be the first of the block change cards; only one is accepted. The format is:

Columns 1 - 4	Card Sequence Number
Columns 5 - 7	'BCH'
Columns 8 - 80	Blank

2. Block change trailer must be the last of the block change cards; only one is accepted. The format is:

Columns 1 - 4	Card Sequence Number
Columns 5 - 7	'BCT'
Columns 8 - 80	Blank

3. Delete

Columns 1 - 4	Card Sequence Number
Columns 5 - 10	'DELETE'
Column 11	Blank
Columns 12 - 80	Term name followed by a colon followed by 'ALL' or any combination of 'SA', 'RT', 'USE' separated by semicolons.

4. Replace

Columns 1 - 4	Card Sequence Number
Columns 5 - 11	'REPLACE'
Column 12	Blank
Columns 13 - 80	Term name to be replaced followed by a colon followed by the replacing name.

An example of two different block changes is given below:

```
0001 BCH
0002DELETE ACUTE POLIOMYELITIS: RT;SA
0003REPLACE ACUTE POLIOMYELITIS: POLIOMYELITIS
0004BCT
```

These cards will result in the deletion of the term ACUTE POLIOMYELITIS wherever it appears in the dictionary file as either a related term (RT) or as a see also (SA) entry. If the term ACUTE POLIOMYELITIS appears other than as a RT or SA entry in any dictionary record, it will be replaced by the single work POLIOMYELITIS, as directed by card 0003 above. Since the DELETE card above eliminated the term from all RT and SA occurrences, the only possible instance of replacement would be in the USE entries.

Note that the block change DELETE does not result in the elimination of the term as a main dictionary entry. The term is deleted only where it is embedded within the data under a main entry. To delete a main dictionary term it is necessary to use the dictionary maintenance cards described below. A main term cannot be deleted from the dictionary by use of the block change.

Dictionary Maintenance Cards

1. HDR card

Columns 1 - 4	Card Sequence Number
Columns 5 - 7	'HDR' (designates the beginning of a set of data pertinent to a term)
Column 8	Blank
Columns 9 - 80	Blank if a new index term is being entered into the dictionary. Term name if maintaining existing dictionary term.

2. ADD cards

Columns 1 - 4	Card Sequence Number
Columns 5 - 7	'ADD' (indicates that a new term is being added to the dictionary or that information is being added to an existing term)
Column 8	Blank
Columns 9 - 80	As described below:
New Term Designation	'NT: Term Name;' Must be present if there was no name on the 'HDR' card. Must be the first of the 'ADD' card entries. Only one NT: is allowed per 'ADD' data set.
Function	PD 'F: (TD;SD;ID);' CD If the term may be used as a descriptor in the master file, either PD, CD, or TD (never more than one) should be entered. A function must

	be given whenever NT: is present. Only one F: ( ) entry is allowed per data set, but may contain more than one entry (e.g., F : (DD;SD;ID)). **PD - Precise Descriptor **TD - Trial Descriptor **CD - Common Descriptor **SD - Subdescriptor **ID - Identification File Entry
Dictionary	'DICT: (DCT;DCT;...);' This entry specifies the dictionaries to which the term belongs. Up to 20 dictionaries may be specified for a term. Only one DICT: entry per 'ADD' data is permitted Each DCT is a 1-3 character entry.
File/Context	'FC: FIL-CXT;' This entry specifies the files in which a term may appear and the context in which it is used within that file. Up to 20 FC: entries may be specified for each main term or related term. Each FIL and CXT is a 1-3 character entry.
See Reference	'USE: Term Name;' This entry specifies a preferred term which is to be substituted for the index term following the NT: entry. Only one USE: may be entered for a term.
Related Term	'RT: Term Name;' Up to 40 related terms may be entered for one index term. None may be entered if 'USE' has been specified.
See Also	'SA: Term Name;' Up to 40 see also terms may be entered for one index term. None may be entered if a 'USE' term was specified.

3. Delete Cards

Columns 1 - 4	Card Sequence Number
Columns 5 - 10	'DELETE' (indicates that the information which follows is to be deleted from an existing dictionary record)
Column 11	Blank
Columns 12 - 80	As described below (DELETE card entries have the same restrictions as do the ADD card entries except that the FC: entries may only follow the main term): If the file/context entries following a related term must be changed, the related term must be deleted and re-entered with the correct file/context data.

Function	PD 'F: (CD;SD;ID):' TD
Dictionary	'DICT: (DCT;DCT;...);'
See Reference	'USE: Term Name;'
Related Term	'RT: Term Name;'
See Also	'SA: Term Name;'
File Context	'FC: FIL-CXT;'
Scope	'SCOPE: ALL or N_1;N_2;...N_m' where N is a 1-3 digit number from 0 - 255. Up to 256 note numbers may be specified. If SCOPE is specified, it must be the last entry in the DELETE data set.

4. Replace cards

Columns 1 - 4	Card Sequence Number
Columns 5 - 11	'REPLACE'
Column 12	Blank
Columns 13 - 80	As described below:

See Reference	'USE" Term to be replaced; Replacing Term;'
Related Term	'RT: Term to be replaced; Replacing Term;'
See Also	'SA: Term to be replaced; Replacing Term;'

5. Scope cards

Columns 1 - 4	Card Sequence Number
Columns 5 - 9	'SCOPE'
Columns 10 - 12	Scope note line number Must be a three digit number 001 - 255 with any necessary leading aeros.
Column 13	Blank
Columns 14 - 73	Scope Note (all characters are valid)
Columns 74 - 80	Blank

An example showing the addition of a new term to the dictionary and the revision of an existing one is given below.

```
0001        HDR
0002        ADD NT: EMPIRIN; F: (PD); DICT:(DRG);
0003            FC: HUM-DRG; RT:ACETYLSALICYLIC ACID
0006        SCOPE040 REF/AMERICAN DRUG INDEX
0007        HDR MEASURIN
0008        ADD DICT: (DRG); RT:NON-NARCOTIC
```

Card 0001 is a header card. It is blank from column 8 because it precedes the addition of a new main term to the dictionary. The term itself is EMPIRIN, shown following the code NT: in the first ADD card after the HDR. Card 0003 is a continuation of the data about EMPIRIN, giving the File/Context code and a related term. A single line of scope note is given for this term as well. Note that the line number need not be 001. It may be any number desired. However, the scope note will be assigned a different number internally so that additional scope notes may be inserted between lines, if desired.

Processing

DEDIT is the buffer between the people who create update data for the master dictionary file and the program (DICT2) which updates the file. As such, it edits and reformats raw data and flags certain erroneous data with messages explaining the nature of the error. The format of the input data is explained in the INPUT section. DEDIT checks each card to make sure that it conforms to the basic input format. If it does, the nature of the modification requested is determined. If the request is valid, records are created for input to the dictionary maintenance program. All cards pertaining to that request are processed. When a new request is encountered, the records thus created are output on tape. Then the new request is processed. Because of the order in which modification data must be submitted to DEDIT, all block changes will be processed first and will be written at the beginning of the output tape. Modifications to individual dictionary entries will follow. When an end-of-file occurs on the input file, processing will be terminated.

Output

DEDIT produces a tape of changes on SYS005. These records, once sorted, provide the input for DICT2. In addition to the tape of valid changes, error messages pertaining to faulty input data are printed along with the data to which they pertain.

DICTIONARY UPDATE PROGRAM (DICT2)

Purpose

The purpose of the dictionary update program is to update the dictionary update file (tape). This program also builds the dictionary file disk.

Input

1. Mount the sorted update tape on SYS006. This tape is output from the dictionary edit program and then sorted on main term.
2. Mount the prior dictionary file tape on SYS007.

Processing

After opening the various files and initializing switches and counters, this program reads up to 20 block change records from the sorted update tape and stores them in core. Each dictionary record is read from the prior dictionary tape. If the update records compare low or equal to the dictionary record then some sort of updating is done (block change, add, delete, or revise) unless there is an error, such as an attempt to delete a non-existent dictionary term. For each add, delete, or change, three or more records are written on the message tape. For example, if a scope note has been added, then four records are written on the message tape; a change message, a message stating that a scope note has been added, the prior dictionary record and the new dictionary record. If there is an error, one or more error messages will be output indicating the type of error.

All unaffected dictionary records are written on the new dictionary tape as well as new additions and error-free changed records. If there is an error in updating, the prior dictionary record is output on the new dictionary tape. When the dictionary disk is being built, each record that is output to the new dictionary tape is also output to the indexed sequential disk file.

Output

1. Mount a scratch tape on SYS009 which will become the updated dictionary file tape.

2. Mount a scratch tape on SYS008 which will contain the changes and errors from the update program. This tape is printed using the dictionary update message program (DICTMSG).

3. Mount the dictionary file disk on SYS010. If it is not necessary to build a dictionary file disk, then in the DOS job control cards include the following: //UPSI 1XXXXXXX . See DOS JCL for a description of this card (User Program Switch).

DICTIONARY MESSAGE PRINTING PROGRAM (DICTMSG)

Purpose

DICTMSG lists the tape of changes and errors produced by the dictionary maintenance program. The dictionary maintenance program processes three types of dictionary modification requests: additions (new records), changes (to subordinate data within existing records), and deletions (of existing records). In addition, errors are detected and noted as to the cause (i.e., invalid main term referenced, etc.). DICTMSG lists the messages created by the dictionary maintenance program and the dictionary record(s) to which they apply.

Input

1. Tape

   DICTMSG requires that the dictionary maintenance change tape be mounted on SYS006.

2. Control Card

   In some cases the listing of only one or two types of messages and their corresponding dictionary records may be desired. When this occurs, request data is inserted on an UPSI card at the beginning of the program. This card sets certain bits in a communication region of the supervisor which may be sensed by the program. Request data has meaning as follows:

```
//UPSI XXXXXXXX
       12345678
```

A 1 in position four indicates an addition record will be printed.
A 1 in position five indicates a change record will be printed.
A 1 in position 7 indicates a delete record will be printed.
A 1 in position eight indicates an error record will be printed.

A zero (0) in any of the above positions indicates that action is not requested.

Processing

Each logical record is identified as either a message or a dictionary record. Messages are classified as "requested" or "non-requested" depending upon the request data described above. If the message is "requested" (i.e., called for by an appropriate punch in the UPSI card), it is printed and a switch is set indicating that the next dictionary record encountered will be listed. When a dictionary record is identified, the switch mentioned above is tested. If it is set, the dictionary record is printed by the subroutine SBRFORM. Processing continues until all logical records have been read. An end-of-file condition terminates DICTMSG.

SBRFORM formats and prints the data in dictionary records read into the calling program. The print format is described in the OUTPUT portion below. Each term or block of subordinate data is identified and titled before printing. The terms are indented and printed under the main term. The main term is double printed (for emphasis) as are the titles for 'USE' terms and 'SEE ALSO' terms. File/context and dictionary numbers are converted to decimal prior to printing. When the end-of-data is encountered, control is returned to the calling program.

Output

Each message will be printed on a separate line. A partial line of asterisks will separate dictionary records. The dictionary records will be printed in the following format:

```
(main term)    (double printed)
   AUTHORITY--(auth.)
   FUNCTION--(fns.)           DICTIONARIES--(dictionary nos.)
   FILE/CONTEXT NUMBERS--
     (f-c, f-c, f-c, f-c, f-c,...)
   USE--      (double printed)   (term to be used in lieu of main term)
    or
   RELATED TERMS--n          (n=no. of related terms)
     (rt1, rt2, rt3, rt4,...
      rtn-1, rtn)
   SEE ALSO--  (double printed)
     (sa1, sa2, sa3,...)
   SCOPE NOTES--
     LINE NO. n (text)
```

A double space will separate the main term with its pertinent data from the subordinate terms included in the record. Also a double space will separate the dictionary record from the messages that reference it. An example of the actual printout of a message and its dictionary record follows:

```
**************************
 ECOTRIN
MAIN TERM HAS BEEN ADDED
ONE OR MORE ADDITIONS AND/OR DELETIONS


ECOTRIN
  FUNCTION-- PD         DICTIONARIES--1
  FILE / CATEGORY NUMBERS--
    1-5

  USE--  ACETYLSALICYLIC ACID
  SCOPE NOTES--
    LINE NO. 1 SS TRADE NAME HUMAN
    LINE NO. 3 SS MFG SMITH KLINE AND FRENCH
    LINE NO. 4 AMERICAN DRUG INDEX  PHYSICIANS DISK REFERENCE
```

DICTIONARY PRINT PROGRAM (DPRINT3)

Purpose

This program enables the user to print selected portions of the dictionary tape. The user may print an entire main term record or any combination of the following parts of the record: authority of nomenclature*, dictionaries, function, file/context codes, scope notes, see also references, related terms, and use terms. He may either print every term on the dictionary tape, or he may print selected terms. These options are controlled by a user-prepared set of control cards.

* This field type is not generally carried now. Coding to handle it remains in the programs.

Input

This program uses input from both magnetic tape and cards.

1. Tape

   The updated dictionary tape is file protected and mounted on the tape unit assigned as SYS007.

2. Cards

   The control (request) cards which control the printing are read into the 2540 card reader assigned as SYSRDR. There are two basic types of request cards.

   a. Dump Card

      The first type, the dump card, allows the user to print every main term record on the dictionary tape. One form of the dump card indicates that each record is to be printed in its entirety. The format of this request card is as follows:

      ```
      DUMP / RECORD
      ```

The other dump card form allows the user to print selected portions of each record on the dictionary tape. The format of this card follows:

```
DUMP / AN, DICT, FN , FC, SCOPE, SA, RT, USE
```

Where 'AN' indicates authority of nomenclature, 'DICT' indicates dictionaries, 'FN' indicates function, 'FC' indicates file/context codes, 'SCOPE' indicates scope notes, 'SA' indicates see also references, 'RT' indicates related terms, and 'USE' indicates use terms. Any one or more of these options may be selected. The options may be listed in any order.

b. Term Cards

The other type of request is the term card. Term cards may be used instead of a dump card to print specific main term records. There must be a term card for each main term record to be printed.

The first form of the term request card will cause the entire main term record that is specified to be printed. The format of this card follows:

```
TERM / ABBOCILLIN ; PRINT / RECORD
```

The second form of the term card will enable the user to print selected portions of the main term that is specified. The format of this term card follows:

```
TERM / ANTI-INFECTIVES:  PRINT / AN, DICT, FN, FC, SCOPE, SA, RT, USE
```

The options are the as those on the dump request card.

There may be up to one hundred term request cards entered. There may be multiple entries for any one main term.

Processing

Each term request causes the dictionary tape to be searched for the designated main term. For a dump request, each dictionary record is considered a match. If no match is found, an error message is printed and the program continues with the next term request. When a match if found, control is transferred to subroutine SBRFORM which prints the record and then returns control.

When all the requests have been processed, the files are closed and the job is terminated.

Output

Output is handled by the subroutine SBRFORM. Dictionary records will be printed in the following format:

```
(main term)  (double printed)

AUTHORITY--(auth.)
FUNCTION--(fn)
DICTIONARIES--(dictionary nos.)
FILE/CONTEXT NUMBERS--(f-c, f-c, f-c, f-c, f-c,...)
USE--(term)  (double printed)   (term to be used in lieu of main term)
 or
RELATED TERMS--n (where n is the no. of related terms in the record)
  (rt1, rt2, rt3, rt4, rt5,...    rtn-1, rtn)
SEE ALSO--(couble printed)
  (sa1, sa2, sa3,...)
SCOPE NOTES--
  LINE NO. n(text)
```

```
ACETYLSALICYLIC ACID
  FUNCTION-- PD         DICTIONARIES--1
  FILE / CATEGORY NUMBERS--
    1-5

  RELATED TERMS-- 5
    ANALGESIC-ANTIPYRETIC, CNS DEPRESSANTS, NERVOUS SYSTEM DRUGS, NON-NARCOTIC·ANALGESIC, SALICYLATES
  SEE ALSO--
    ECOTRIN, EMPIRIN, MEASURIN, SALACETIN
  SCOPE NOTES--
    LINE NC. 1 SS GENERIC NAME HUMAN
    LINE NO. 4 MERCK INDEX  NATIONAL FORMULARY

HELICON
  FUNCTION-- PD         DICTIONARIES--1
  FILE / CATEGORY NUMBERS--
    1-5

  USE--  ACETYLSALICYLIC ACID
  SCOPE NOTES--
    LINE NO. 1 SS GN-TN
    LINE NO. 2 SS GN-TN
    LINE NO. 4 SS MFG
    LINE NO. 4 REF MERCK INDEX

BARDASE VETERINARY
  FUNCTION-- PD         DICTIONARIES--1
  FILE / CATEGORY NUMBERS--
    2-5

  RELATED TERMS-- 14
    ASPERGILLUS ORYZAE ENZYME, ATROPINE SULFATE, AUTONOMICS, BARBITURATES, CARHOHYDRASE, CNS DEPRESSANTS,
    ENZYME-ENZYME INHIBITORS, HYOSCYAMINE SULFATE, NERVOUS SYSTEM DRUGS, PARASYMPATHOLYTIC, PHENOBARBITAL
    SCOPOLAMINE HYDROBROMIDE, SEDATIVE-HYPNOTIC, TROPANE ALKALOIDS
  SCOPE NOTES--
    LINE NO. 1 SS GN-TN
    LINE NO. 2 SS GN-TN TRADE NAME VETERINARY
```

SYNONYM TAPE PROGRAM (SYNTAPE)

Purpose

The master dictionary tape produced by the DICT2 program contains, interspersed with normal entries, a number of "use" entries, which contain synonymous terms (e.g., "ASPIRIN use ACETYLSALICYLIC ACID").

The SYNTAPE (synonym tape) program examines the master dictionary tape to extract these "use" entries and invert their form, so that the term referred to by the synonymous word appears first in the record. After extraction and inversion, the "use" entries are sorted in sequence by the term referred to, thus bringing together all the variants which for example, will be arranged alphabetically under that term. This sorted tape, used in conjunction with the master dictionary file in DPRINT4 will permit the printing of a dictionary in which all such synonyms are given under the term to which they refer as well as appearing in their normal alphabetic sequence.

Input

The tape created by the dictionary maintenance program is mounted on SYS007.

Processing

A record is read and the contents position is tested for the presence of a "use" term in the record. If there is none, the next record is read. If the contents position indicates there is a "use" term, it is located and written on tape, followed by its length, its main term, and the length of the main term. Then the next record is read. Processing continues until the end-of-file.

Output

Output from SYNTAPE is written on a tape mounted on SYS008. This tape, when sorted, will be used as synonym input to DPRINT4.

DICTIONARY PRINT PROGRAM (DPRINT4)

Purpose

This program enables the user to print the master dictionary tape in either a publishable or a working copy format. The publishable copy is printed as double-length pages with two columns per page. The working copy is printed on single pages with two columns per page.

The user also may specify printing of the entire main term record or any combination of the following parts of the record: authority of nomenclature, dictionaries, function, file/context codes, scope notes, see also references, related terms, and use terms.

In addition, he may select the specific dictionaries to be printed (e.g., he may choose to print only those terms belonging to the Adverse Reactions Dictionary, Legal Dictionary, etc.).

Input

This program requires input from both magnetic tape and cards.

1. Tapes
   - a. The updated dictionary tape is file protected and mounted on the tape unit assigned as SYS007.
   - b. The sorted synonym tape is file protected and mounted on the tape unit assigned as SYS006.
2. Control Cards

   Program control cards are read into the 2540 card reader assigned as SYSRDR. There are three control cards required by this program.

   - a. Copy Control Card

     The first card, the copy control card, allows the user to print the dictionary in either of the two possible formats. The formats of this control card as as follows:

```
COPY / PUBLISH

COPY / WORKING
```

b. Dump Control Card

The second card, the dump control card, allows the user to dump either the complete record for each main term or selected portions of each main term. Following are the formats of this card:

```
DUMP / RECORD

DUMP / AN, DICT, FN , FC, SCOPE, SA, RT, USE
```

When dumping selected portions of the dictionary, any combination of the options below may be selected and in any order.

OPTION	DESCRIPTION
AN	Authority of Nomenclature
DICT	Dictionaries
FN	Function
FC	File/Context Numbers
SCOPE	Scope Notes
SA	See Also References
RT	Related Terms
USE	Use Terms

c. List Control Card

The list control card allows the user to select the dictionaries to be dumped. He may either specify that all dictionaries should be included in the printed report or he may list the dictionary mnemonics of those dictionaries to be printed. The formats of this card are as follows:

```
LIST / ALL

LIST / XXX, XXX, XXX, XXX, ..., XXX
```

Where the XXX indicates a three character dictionary mnemonic, as many dictionaries are allowed as will fit on one card.

The spacing on these cards is unrestricted; blanks between words are ignored.

Note the distinction between the functions of the list control card and the dump control card. The former controls which dictionary terms are to be printed. The latter controls the amount of data to be printed under each term.

d. Header Control Card

The header control card contains up to 60 characters of data which will be used as a header for each page in the dictionary. The format is simply the word HEADER followed by a slash (/) followed by the header data.

e. Restart Control Card

The restart card allows an operator to begin printing in the middle of the dictionary. It is formatted as follows:

```
RESTART / PAGE=NNN, TERM=XXXXX

RESTART / TERM=XXXXX
```

Note that the page number is optional. If not included, pagination will begin at one. The term parameter must be included. Both the dictionary and synonym tapes will be positioned to begin at the specified term.

Processing

After initializing all switches, counters and working areas, and opening the files, this program begins reading cards, searching for one each of the three necessary kinds of control cards (copy control card, dump control card, and list control card). It will continue reading cards until a system end-of-file card is encountered. The program will accept the first three valid types it encounters, in any order. This means that there may be any number of cards in the reader preceding the system end-of-file. The program will simply accept the first three valid control cards and ignore the remainder. If the program does not find three valid cards, the job is terminated.

Once the printing parameters have been determined, the program reads records from the master dictionary tape and the sorted synonym tape (which was produced by the SYNTAPE program). Dictionary terms which meet the criteria specified in the list control card are printed along with whatever elements of data were specified by the dump control card and with the corresponding entries from the synonym tape, if any. If a given dictionary term is not to be printed, it is ignored (along with its associated synonyms, if any) and the next dictionary term is inspected.

This process continues until end-of-file is reached on the dictionary tape. If, during the printing process, any records are encountered on the synonym tape which do not correspond to entries on the dictionary tape, there is an editorial error in the dictionary. Existence of a record on the synonym tape in the form 'XXX use(d for) YYY', for example, means that there must be a record on the dictionary tape in the form 'YYY use XXX'. The editorial rules require that for such "use" entries, 'XXX' be a valid dictionary term. The failure of the record 'XXX' use(d for) YYY' on the synonym tape to match with any entry on the dictionary tape means that XXX does not exist as a main term in the dictionary. Therefore, the cross reference, 'YYY use XXX' is invalid.

At the conclusion of the printing of the dictionary, these unmatched synonym records are printed for inspection so that the dictionary editorial staff may make corrections.

It is possible to execute this print program without having a synonym tape available. This is done my mounting a scratch tape which has three tape marks written at its beginning in lieu of the synonym tape on SYS006. This will result in printing of the dictionary master tape but, of course, without any 'used for' entries under any term.

Output

DPRINT4 will produce one of two types of printer output. These will vary only in the length of the page (i.e., if a publishable copy is requested, the page length will be equal to two printer pages. For a working copy, only one page will be printed). Both types of output will follow the format below:

```
(main term)
AUTHORITY--(auth.)
FUNCTION--(fns.)  DICTIONARIES--(d1, d2,...)
FILE / CONTEXT NUMBERS--(f-c, f-c, f-c, f-c,...)
USE--(term to be used in lieu of main term)
  or
RELATED TERMS--
  (rt1     rt2
   rt3     rt4
   ...)
SEE ALSO--
  (sa1     sa2
   sa3     sa4)
USED FOR--
  (synonym1          synonym2
   synonym3          ........)
SCOPE NOTES--
  XXXXXXXXXXX
  XXXXXXXXXXX
```

Each page will be divided into two columns, each following the above format. If a main term record is split between columns (i.e., if the entire record will not fit one column), a note will precede the continuation in the second column. For example, ASPIRIN (cont'd.)

```
CALCIUM QUINATE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERM --
    MINERALS(1-2)

CALCIUM SACCHARATE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERM --
    PHARMACEUTICAL ADJUN(1-2)

CALCIUM SACCHARIN
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERM --
    SALT-SUGAR SUBSTITUT(1-2)

CALCIUM SALICYLATE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    ANTIDIARRHEA(1-2)                  SALICYLATES(1-4)

CALCIUM SUCCINATE.
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERM --
    ANALGESICS-ANTIPYRETIC(1-2)

CALCIUM SULFATE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    INORGANICS(1-4)
    PHARMACEUTICAL ADJUN(1-2)

CALCIUM SULFIDE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    ANTISEPTICS(1-2)                   INORGANICS(1-4)

CALCIUM THIOSULFATE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERM --
    ANTI-INFECTIVES(1-2)

CALCIUM TRISODIUM PENTETATE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERM --
    HEAVY METAL ANTAGONI(1-2)

CALCIUM UNDECYLENATE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    ANTI-INFECTIVES(1-2)
    ANTIFUNGAL DRUGS(1-2)

CALCIWAFERS
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    CALCIUM GLUCONATE(1-3)
    CALCIUM PHOSPHATE DI(1-3)     ERGOCALCIFEROL(1-3)
    FAT SOLUBLE VITAMINS(1-2)     INORGANICS(1-4)
    VITAMIN D(1-2)
    VITAMIN-MINERAL PREPARATION(1-2)

CALCOTHEOBROMINE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    CARDIOVASCULAR DRUGS(1-2)     DIURETICS(1-2)
    SALICYLATES(1-4)              XANTHINES(1-4)

CALCREOSE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    ANTISEPTICS(1-2)                   EXPECTORANTS(1-2)

CALCRETIN
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1
```

```
CALCRETIN(CONT.)

  RELATED TERMS--
    BARBITURATES(1-3)                  ENZYMES(1-2)
    ETHYL AMINOBENZOATE(1-3)           PANCREATIN(1-3)
    PAPAIN(1-3)                        PHENOBARBITAL(1-3)
    SEDATIVES-HYPNOTICS(1-2)

CALDECORT
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    ADRENAL CORTICOSTEROID(1-2)   ANTIBIOTICS(1-2)
    ANTIFUNGAL DRUGS(1-2)
    CALCIUM UNDECYLENATE(1-3)
    HYDROCORTISONE ACETA(1-3)
    NEOMYCIN SULFATE(1-3)         STEROIDS(1-4)

CALDESENE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    ANTI-INFECTIVES(1-2)
    ANTIFUNGAL DRUGS(1-2)

CALENDULA
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERM --
    ANTISEPTICS(1-2)

CALFEVITA
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    CALCIUM PHOSPHATE DI(1-3)     ERGOCALCIFEROL(1-3)
    FERROUS SULFATE(1-3)          INORGANICS(1-4)
    IRON PREPARATIONS(1-2)
    THIAMINE HYDROCHLORI(1-3)     VITAMIN A(1-2)
    VITAMIN A PALMITATE(1-3)
    VITAMIN B COMPLEX(1-2)        VITAMIN D(1-2)
    VITAMIN-MINERAL PREPARATION(1-2)

CALGAGE-S
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERM --
    DIAGNOSTIC AIDS(1-2)

CALGLUCON
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    METABOLIC AGENTS(1-2)         MINERALS(1-2)
    REPLACEMENT SOLUTION(1-2)

CALHYDRATE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    CALCIUM GLYCEROPHOSH(1-3)     CALCIUM LACTATE(1-3)
    DIURETICS(1-2)                MERCURIALS(1-4)
    MERSALYL(1-3)                 MINERALS(1-2)
    RENAL AGENTS(1-2)             THEOPHYLLINE(1-3)
    XANTHINES(1-4)

CALIGESIC
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    ANTI-INFLAMMATORY AGENT(1-2)
    CALAMINE(1-3)
    ETHYL AMINOBENZOATE(1-3)
    LOCAL ANESTHETICS(1-2)
    M-CRESOL HEXYLATED(1-3)

CALINESE
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1

  RELATED TERMS--
    ANTACID-ADSORB-DEMUL(1-2)
    CALCIUM CARBONATE(1-3)
    GASTRO-INTESTINAL AGENT(1-2)
    INORGANICS(1-4)
    MAGNESIUM CARBONATE(1-3)
    MAGNESIUM TRISILICAT(1-3)

CALIOBEN
  FUNCTION-- PD,SD       DICTIONARIES--1
  FILE / CONTEXT CODES--
    1-1
```

SUBORDINATE TERM EXTRACTION (RELTERM)

Purpose

When terms are entered into the dictionary or when alterations are made to the dictionary, some types of error checking are not feasible. For example, it would be extremely time consuming to scan the entire dictionary each time a main term is altered or deleted. Such a scan would ensure that the modification will not leave some subordinate term without an entry as a main term. RELTERM accomplishes this purpose. It extracts all subordinate terms of a particular type (e.g., related term, use term, or see also term) and passes them to RELTRM1 which validates each term.

Input

1. The dictionary tape is mounted on SYS007.

2. A UPSI control card determines which types of terms will be extracted. Its format is:

```
//UPSI 00000xxx
             SRU
```

A 1 in the S position causes see also terms to be extracted.
A 1 in the R position causes related terms to be extracted.
A 1 in the U position causes use terms to be extracted.

If no control card is included, related terms will be extracted.

Processing

RELTERM scans each record to determine whether or not the requested term types are present. If they are, each term selected is written out on tape along with the main term to which it refers. Then the next dictionary record is read. An end-of-file on the dictionary tape causes processing to terminate.

Output

The terms extracted and their associated main terms are written on a tape mounted on SYS005. This tape, after sorting will provide the input to RELTRM1.

SUBORDINATE TERM VALIDATION PROGRAM (RELTRM1)

Purpose

After the subordinate terms have been extracted and sorted, RELTRM1 performs a validation against the dictionary. Any term not in the dictionary as a main term will be detected and listed for correction.

Input

1. The dictionary tape is mounted on SYS007.

2. The sorted extracted subordinate terms will be on a tape mounted on SYS005.

Processing

RELTRM1 processes each set of subordinate terms separately. That is, all of the use terms are validated first (if there are any), then related terms (if any have been extracted), and finally see also terms (if any) are validated. The dictionary tape is rewound for each set of terms checked. The sorted extracted terms are compared to the dictionary main terms. If an extracted term is not located as a main term in the dictionary, that subordinate term and the main term with which it was found are written on the work tape. After the entire tape of extracted terms has been processed, the work tape is rewound and printed.

Output

1. A work tape used to contain missing terms is mounted on SYS004. This tape is later printed by RELTRM1.

2. The printed listing of all subordinate terms which were not found as main term entries in the dictionary. An example of the listing is shown below.

AQUADIAL WITH TESTOSTERONE
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
AQUADIOL W/ TESTOSTERONE

BECOTIN WITH C
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
BECOTIN W/ C

BENANSERNIN HYDROCHLORIDE
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
BAS
SEROTONIN BENZYL ANALOG

CAFFEINE AND SODIUM BENZOATE
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
CAFFEINE SODIO-BENZOATE
CAFFEINE SODIUM BENZOATE

CARDAMON SEED
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
GRAINS OF PARADISE

CHIMAPHILA
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
BITTER WINTERGREEN
GROUND HOLLY
PINE TULIP
PIPSISSEWA
PRINCES PINE
PYROLA
RHEUMATISM WEED

COMBEX WITH VITAMIN C
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
COMBEX W/ VITAMIN C

DEXTROMETHROPHAN HYDROBROMIDE
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
D-METHORPHAN HYDROBROMIDE

EN-CEBRIN WITH FLUORIDE
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
EN-CEBRIN W/ FLUORIDE

ERGOAPIOL WITH SAVIN
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
ERGOAPIOL W/ SAVIN

GOODYS HEADACHE POWDERS
OCCURS WITH THE FOLLOWING MAIN TERMS AS A USE TERM.
GOODYS POWDERS

THE MAIN FILE SYSTEM

FILE MAINTENANCE SYSTEM

The heart of the FDA information retrieval system is the series of programs known collectively as the file maintenance system. It accepts and processes the data from the documents to build a variety of information files which are capable of being searched by the retrieval programs.

The Files

There are four closely inter-related, but physically distinct, files of information involved in the file maintenance system. The most significant of these is the master file. This file contains all information from which the others are built, and is the basis of the retrieval system. The contents of this file are described in detail elsewhere, but are summarized here for reference. The master file contains a series of individual records, one for each document which has been entered into the system. Each individual record is divided into two portions. The searchable section contains descriptors, subdescriptors, numeric values, and similar data. It is that portion of the record which is used for the actual search process. The second portion of the record is known as "free text". Free text can be, as the name implies, any kind of information one wishes to put into the record. It can be an abstract, a summary of the original document, doctor's comments, or whatever is desired. The free text can be treated as a series of "paragraphs" or "segments" if desired. Different kinds of information may be recorded in different segments. For example, one may put the hospital identification data in segment 1, diagnostic information in segment 2, laboratory test data in segment 3, and so forth, depending upon the user's requirements.

The master file is maintained in the IBM 2321 data cell (or its equivalent), a large-capacity direct access storage device. This permits immediate access to any specific record in the entire file, analogous to the manner in which one may turn to a particular page of a book without having to peruse the preceding pages.

The records are maintained in this storage device in sequence by an "internal document number" which is used solely for convenience in computer processing. Each new document, as it is added to the file, is assigned the next sequential internal document number. Thus, all documents added to the master file are automatically stored after previous records.

In order to permit easy reference to the machine-generated internal number, a separate file (in effect a dictionary) is maintained. This file contains cross references from the user's external document number (which may be alphabetic, numeric, or both, up to eight characters in length) to the machine-generated internal number. It is called the cross reference file.

A third file, called the inverted file, serves as a detailed index to the master file. While the cross reference file described above provides a single entry for each document (i.e., one external document equated to one internal number), the inverted file provides multiple references to a given document. Each term (descriptor) used to describe a document is entered in this inverted file, alphabetically with other terms. Every term is followed by a list of document numbers (internal numbers in this case) which identify the documents to which the term has been applied. Under the term PENTOBARBITAL SODIUM, for example, will be the list of documents to which that term has been applied.

Certain very common terms, such as DATE OF BIRTH, and the like, are not indexed in this manner in the inverted file, since to do so would create an excessively large volume of data. The use of these "common descriptors" is described elsewhere. The inverted file, like the cross reference file is kept on a high-speed direct access storage device such as the IBM 2311.

The fourth file in this system is the ID (identification) file. This is a special purpose file which contains, in effect, a subset of the master file. It is written on magnetic tape (rather than on a direct access device) and is in the same format. It contains user-selected portions of the searchable section of the record for every document, but contains no free text. It is used for certain types of searches which are not feasible with the master file.

These four files represent the complete data base for the entire information retrieval system:

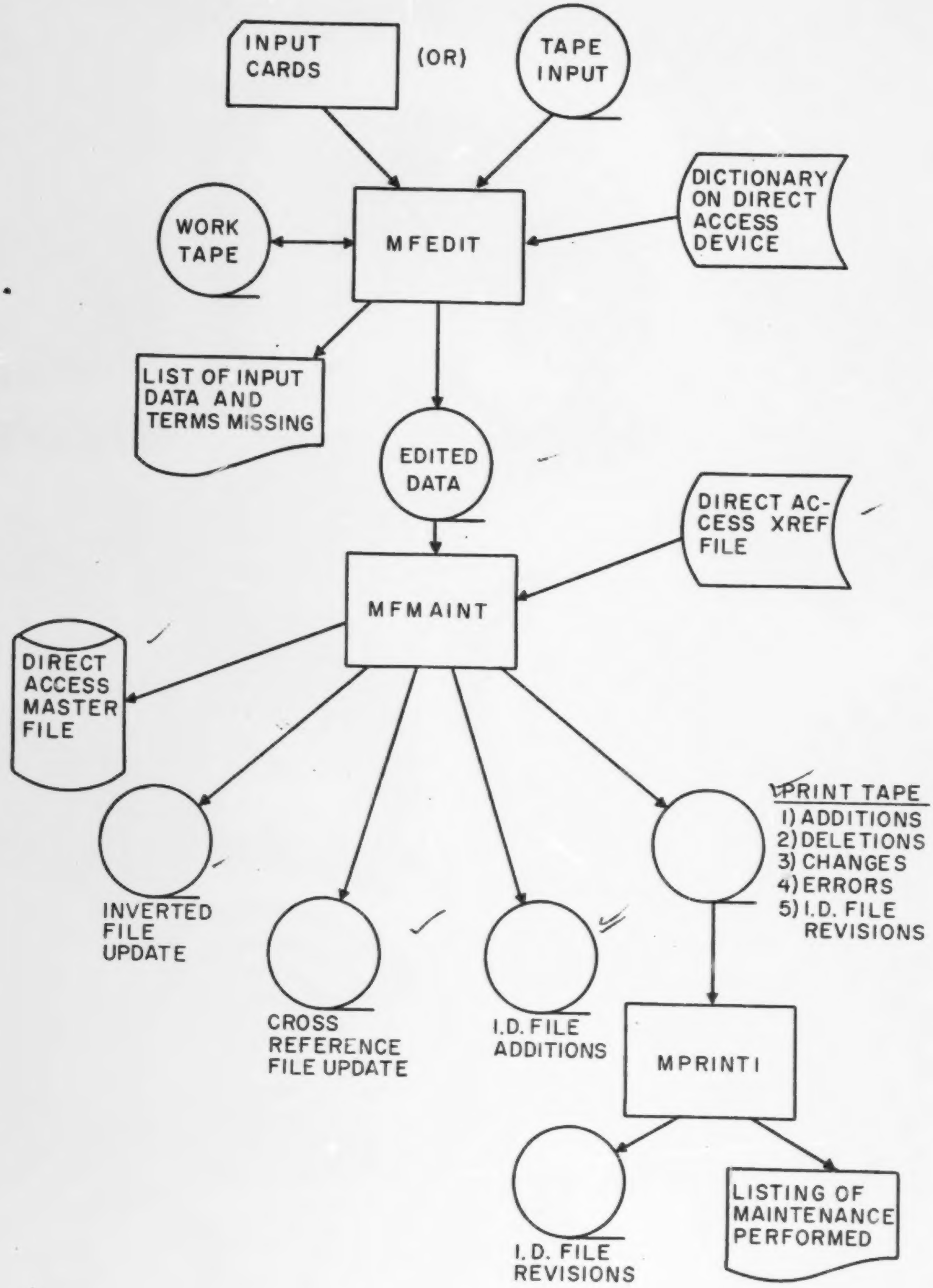
1. Master File

   The master file contains all information, both descriptive terms and free text, pertaining to all documents stored in the system.

2. Inverted File

   The inverted file is a subject index to the master file.

MASTER FILE MAINTENANCE



3. Cross Reference File

The cross reference file relates user assigned document numbers to the document numbers assigned by the system.

4. Identification File

The identification file is a condensed form of the master file which typically contains the common descriptor portions of the master file records plus terms specifically tagged to become ID file entries.

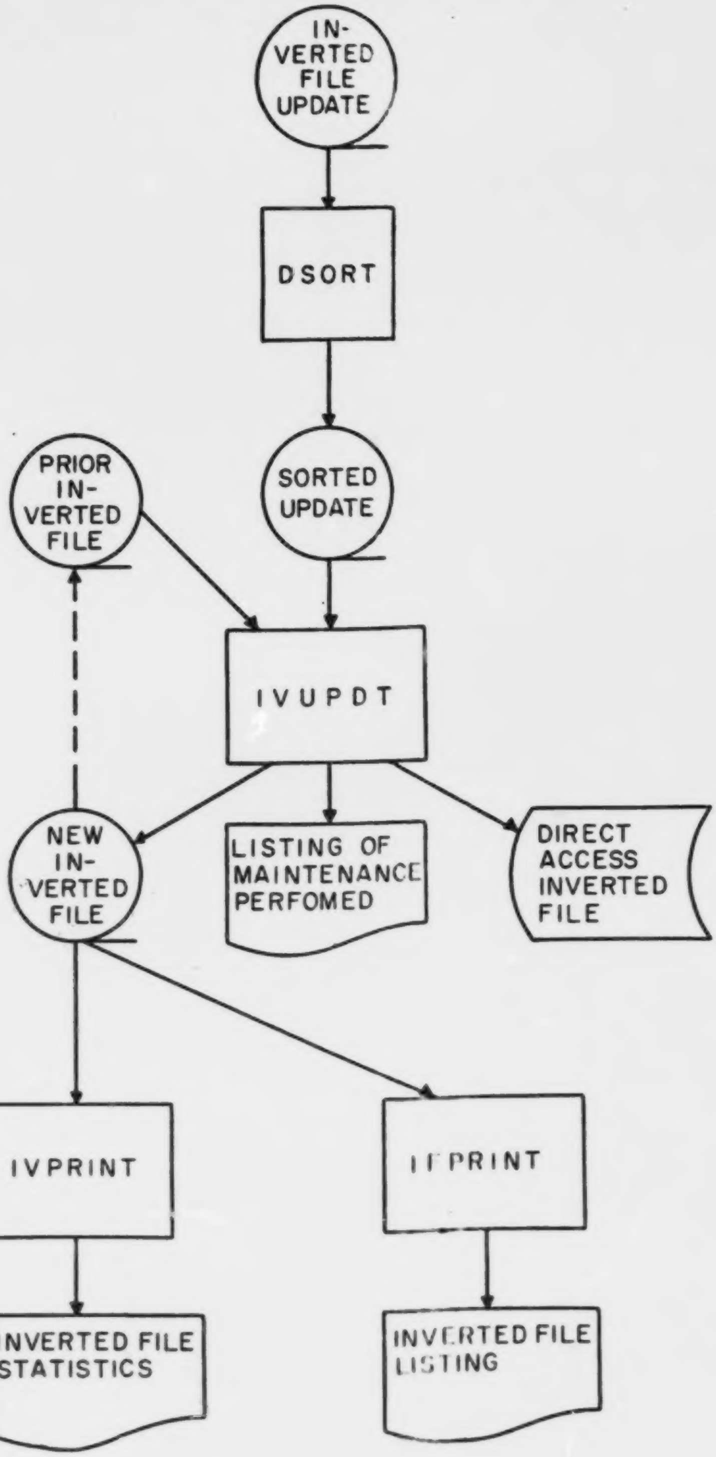
The programs which create these files are inter-related in such a way as to ensure that, short of a deliberate stoppage, the files will always remain "in phase". That is, the programs, if allowed to run to completion, will always update all four files in such a way that any change in the master file will be correctly reflected in the other three. It is not possible to change any of the other three without first changing the master file.

The Programs

A series of nine programs comprises this file maintenance system. They are: MFEDIT, MFMAINT, UPDXREF, IDUPDATE, IFUPDATE, MPRINT1, EDITBC, DCELLDMP, and LOADCELL. Like the dictionary system, two of these perform the basic functions, while the remainder are, in effect, "satellite" programs.

The two fundamental programs are MFEDIT and MFMAINT, master file edit and master file maintenance, respectively. MFEDIT accepts card (or card images on tape) input of additions, changes, and deletions to the master file. All the various elements of information, such as the descriptors, subdescriptors, and the like, are validated by checking them against the disk file version of the dictionary. In addition, a large number of internal consistency checks are made, ensuring, for example, that every subdescriptor is in fact associated with some descriptor. MFEDIT is designed in such a way that, if any portion of the record for a given document is found in error, the entire document is rejected. This ensures that only logically complete documents are passed onto the master file maintenance program.

INVERTED FILE MAINTENANCE



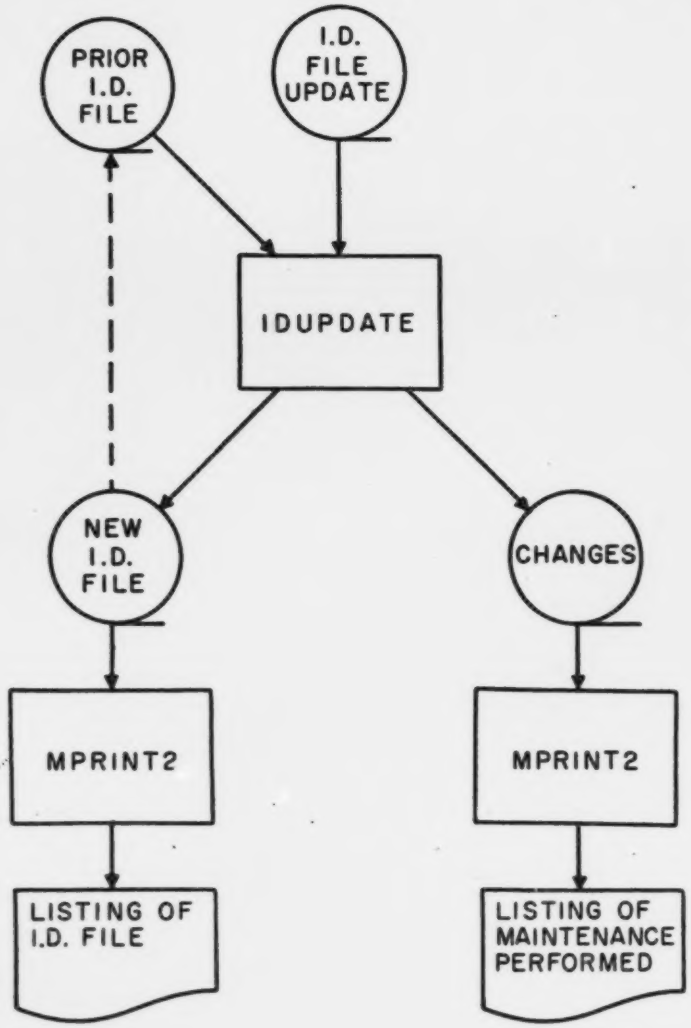
That program, MFMAINT accepts the validated data from MFEDIT, performs the operations indicated (e.g., adds the new document, revises an existing document by adding or deleting elements of information, or deletes a document from the file). Additional editing is performed by this program. It prints error messages if one attempts to delete a nonexistent document, and the like. MFMAINT makes the changes directly to the master file maintained in the data cell, and produces tape records required to alter the status of the other files to bring them in correspondence with the master file. These tapes are the input to the three auxiliary file maintenance programs UPDXREF, IDUPDATE, and IFUPDATE (update cross reference file, identification file update, and inverted file update).

All three of these latter programs operate in essentially the same manner. The new information is read in conjunction with the prior version of the appropriate file and the indicated changes are made, thus producing a current version of that file. In the case of the cross reference and inverted files, both a tape and a disk version of the updated file are produced. For the identification file only a tape is produced.

The continued updating of the master file in the data cell will ultimately result in the need for re-organization of the information in that device and the necessity to generate back up on magnetic tape. Programs are therefore provided to re-organize the data cell. DCELLDMP (data cell dump) is a program which will read all information from the data cell, in ascending sequence by internal document number, and copy it onto tape. LOADCELL reverses the process, reading from the tape and loading the data cell sequentially. This process (DCELLDMP followed by LOADCELL) will optimize the file organization of the data cell.

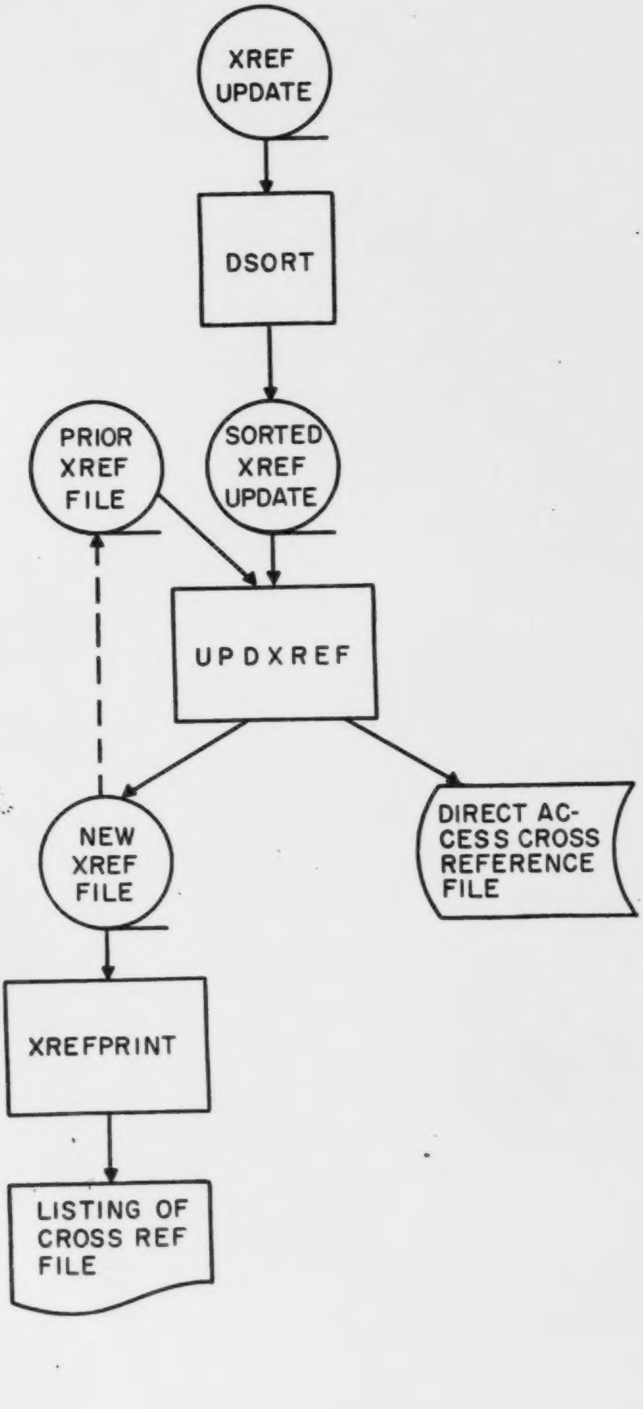
Since, in the process of executing DCELLDMP, every record in the file must be read sequentially, there is an opportunity to make certain changes to those records. It may be desirable, for example, to change every occurrence of the descriptor ATOMIC PILE to NUCLEAR REACTOR. It may be desirable simply to delete every occurrence of a given term. This facility is provided by the block change feature of DCELLDMP. It is possible to prepare cards to perform a delete or replace operation, if desired. The program EDITBC (edit block changes) validates these desired changes against the dictionary disk file and stores them for use by DCELLDMP. As the latter program is reading the records out of the data cell onto tape, it consults the list of block changes and alters the master file records accordingly. Thus, the resultant tape file, at the conclusion of DCELLDMP, is a completely current version of the master file. In addition, the program generates tapes which will serve as input to the inverted file and identification file update programs so that those files will also reflect the current status of the master file.

I.D. FILE MAINTENANCE



The last program in the file maintenance system series is MPRINT1, a program which can be used to print master file records from a variety of sources. It is normally used to print the changes, additions, and deletions at the conclusion of a MFMAINT run, but it is also embedded within LOADCELL to permit the printing of the entire file, if desired, when reloading the data cell from tape.

CROSS REFERENCE FILE MAINTENANCE



FILE MAINTENANCE INPUT LANGUAGE

The file maintenance language uses a free form format to ease the burden of indexing and key punching the data. The types of data recognized by the file maintenance system are:

1. Document Number
2. Document Title
3. Descriptors and their Context Codes (precise, common, trial, and unverified)
4. Subdescriptors
5. Numeric Values
    a. Single Values
    b. Lower - Upper Values
    c. Time Values
    d. Date Values
6. Links
7. Free Text

The above data is entered into four types of card records:

1. Header
2. Title
3. Add
4. Delete

The contents of the cards vary somewhat depending upon whether data is being added, altered, or deleted.

Header

Columns 1 - 5	Card Sequence Number
Columns 6 - 8	'HDR'
Column 9	Blank
Columns 10 - 80	Header Card Data

Data pertaining to each document must be prefaced by a header card. The data entered in the header card determines what operations are to be performed. Possible actions to be taken are:

1. ADD
2. REV(ise)
3. DEL(ete)

There are two document numbers associated with every record in the master file. They are:

1. External Document Number - nine characters maximum
2. Internal Document Number - 256^3 -1 maximum

The external number is the one assigned by FDA for the purpose of logging, filing, etc. The internal document number is assigned by the file maintenance program for use in the system's internal directories. The external number may be alpha-numeric while the internal number is a sequentially assigned binary number.

Title

Columns 1 - 5	Card Sequence Number
Columns 6 - 10	'TITLE'
Column 11	Blank
Column 12	Document Title - maximum of 24 characters

There are no restrictions, other than the maximum length, on the title data. Since its primary purpose is for printing, it should be composed of common computer printer graphics.

Add

Columns 1 - 5	Card Sequence Number
Columns 6 - 8	'ADD'
Column 9	Blank
Column 10	Data to be added in free form indexing language. Descriptors, subdescriptors, numeric values, data values, time values, and links are entered via the add card. Free text is also entered using the add card, but it represents a special case in formatting.

Delete

The delete card makes possible the deletion of an entire document, descriptors, subdescriptors, values, links, or free text.

A more detailed description of the file maintenance language follows:

1 - Header Card

Columns 1 - 5	Card Sequence Number
Columns 6 - 8	'HDR'
Column 9	Blank
Columns 10 - 80	ADD : EXTERNAL REV : INTERNAL DEL : INTERNAL ; Security : No Yes ; XOR : YES

ADD: EXTERNAL

When a document is being entered into the information retrieval system, the HDR card must contain the ADD : EXTERNAL entry. If, for example, a document numbered FDA 1234 were to be entered into the master file, the ADD entry would appear as ADD : FDA 1234. Once entered into the master file it will be assigned an internal number, but its identity as FDA 1234 will be maintained.

REV : INTERNAL

When making changes to a document already in the master file, the record must be referred to by the number assigned to it by the system. This number is referred to as the INTERNAL number. If the system assigned the number 60751 to FDA 1234, and we wish to revise the document's contents, we must enter REV : 60751. A cross reference between external and internal numbers is maintained by the system.

DEL : INTERNAL

To delete a document from the master file, the user merely makes one entry: DEL : INTERNAL document number. This entry results in the deletion of all data pertaining to a document (all searchable terms and free text). Deletion of selected portions of a document are considered revisions.

SECURITY : $\begin{matrix}\text{NO}\\\text{YES}\end{matrix}$

The SECURITY entry is optional. In the event one wishes to classify a document, the security entry must be used. When used, the search program will omit classified documents unless they are specifically requested. That is, a classified document search is run.

XOR : YES

Before a document can be entered into the system, it is looked up in the cross reference file. The document is rejected if it is already present in the cross reference file. To override this restriction the cross reference override entry in conjunction with ADD : EXTERNAL must be used.

2 - Title Card

Columns 1 - 5	Card Sequence Number
Columns 6 - 10	'TITLE'
Column 11	Blank
Columns 12 - 80	Title Information

Title

Up to twenty four characters of title information may be stored with a document. This data is not searchable and is only used for printing.

3 - Add Card

Columns 1 - 5	Card Sequence Number
Columns 6 - 8	'ADD'
Column 9	Blank
Columns 10 - 80	D : DESCRIPTOR ; C:CXT ; LK:(1,2,...16) ; SUBDESCRIPTOR LK:(1,2,...16) ;

various forms of values include:

SV : DEC. or
LV : FLOATING PT
UV : NUMBER
DV : MM-DD-YY
TV : HH.MM.SS

D : DESCRIPTOR

A descriptor may consist of from one to sixty four alphabetic or numeric characters. A few graphics may not be used in forming descriptors as they serve as punctuation in the indexing language. The reserved characters are: semicolon and colon.

C : CXT

The CXT entry is provided to allow the user to indicate the context in which a descriptor is used. In one case a rash may be a symptom while in another case it might be a reaction. The distinction can be made by using the CXT code. The code itself is an abbreviation consisting of from one to three alphabetic characters. The codes are stored as a single byte binary number. Hence; there is provision for 256 unique context codes for each data file in the system. In addition to the CXT entry in the indexing language, provision has been made to append additional context codes during the dictionary validation phase of the file maintenance system.

LK : (1,2,...16)

The information retrieval system makes provision to relate a descriptor to one or more descriptors by means of the LK (links) entry. There are sixteen possible link entries for a given descriptor.

S : SUBDESCRIPTOR

The subdescriptor provision allows the indexer to refine or qualify the meaning of a descriptor. Mass, for example, may be in either grams or kilograms. The subdescriptor provides the ability to draw such a distinction. The information retrieval system also makes provision to relate a subdescriptor to one or more subdescriptors by means of the LK (links) entry. There are sixteen possible link entries for a given subdescriptor.

SV : VALUE, LV : VALUE , UV : VALUE

Of the five types of values recognized by the system, three of them represent measures of various types of quantities and are entered in the same format. Numbers can be entered in either normal decimal notation such as the number 151.09 or floating point notation. The floating point notation is a computer oriented form of scientific or what is commonly referred to as engineering notation. It consists of a characteristic and a mantissa in the reverse order. The number 1234.56 can be expressed as .123456 x 10^4. In floating point notation, it would be expressed as .123456+E4. In general, the format of a floating point number is:

> ± D.DDD E ± nn where D represents a decimal digit and n represents the power of ten which the decimal point must be moved to express the value being represented. Tolerances may be indexed by using the lower-upper value provision of the system. If used, both the lower and upper limits must be specified.

DV : MM-DD-YY or DV : MM-DD-YYYY

Dates are indexed via the DV or date value entry. The system treats dates differently than ordinary numbers since the year must precede the month and day to effect accurate comparisons. If the YY format is used, the system assumes 19YY. For dates outside the twentieth century, all four digits of the year must be entered.

TV : HH.MM.SS

Time of day may be entered via the TV provision. HH represents hour, MM minutes, and SS for seconds.

Add Text Card

Columns 1 - 5	Card Sequence Number
Columns 6 - 8	'ADD'
Column 9	Blank
Columns 10 - 80	T : SEGNO

T : SEGNO

This entry precedes the text data and is on a card by itself. SEGNO may vary between 1 and 255. The text follows in the ensuing cards. Text starts in column ten and can run as far as column eighty. There are no restrictions on the character set entered into the free text data. The format for the text card is:

Columns 1 - 5	Card Sequence Number
Columns 6 - 9	Blank
Columns 10 - 80	Text Data

Delete Card

Columns 1 - 5	Card Sequence Number
Columns 6 - 11	'DELETE'
Column 12	D (descriptor) S (subdescriptor) V (value) L (link or links) T (text segment)
Column 13	Blank
Columns 14 - 80	D : DESCRIPTOR ; S : SUBDESCRIPTOR ; T : SEGNO various forms of values include: SV : DEC. or LV : FLOATING PT UV : DV : MM-DD-YY TV : HH.MM.SS

Delete Descriptor

To delete a descriptor a 'D' is punched in column 12 and the D : DESCRIPTOR entry is used. When a descriptor is deleted, its subdescriptors and/or numeric values, and links are also deleted.

Delete Subdescriptor

To delete a subdescriptor, a 'S' is punched in column 12. In this case, both the D : DESCRIPTOR and the S : SUBDESCRIPTOR entries must be present. The descriptor entry in this case informs the system under which descriptor the subdescriptor to be deleted is stored. When a subdescriptor is deleted, its values and links are also deleted.

Delete Value

Values may pertain to a subdescriptor or a descriptor, depending upon whether or not a subdescriptor is present. If a value is stored under a subdescriptor, it pertains to the subdescriptor. If it is not stored under a subdescriptor, it pertains to the descriptor. In deleting a value associated with subdescriptors, which is the usual case, both the D : DESCRIPTOR ; and S : SUBDESCRIPTOR ; entries must be present as well as the value to be deleted.

When deleting a value not associated with a subdescriptor, only the D : DESCRIPTOR entry is needed prior to the value to be deleted. When deleting a lower-upper pair of values, both the lower (LV) and the upper (UV) must be present.

When a descriptor is used to 'locate' a subdescriptor within a record for the purpose of deleting (or adding) that subdescriptor, the descriptor itself is not modified. It is referred to as a 'locator' descriptor in the MFMAINT program. 'Locator' subdescriptors serve a similar function in deleting or adding values.

Delete Links

Links can be deleted from a descriptor and/or subdescriptor in the same manner as values.

MASTER FILE ORGANIZATION

Introduction

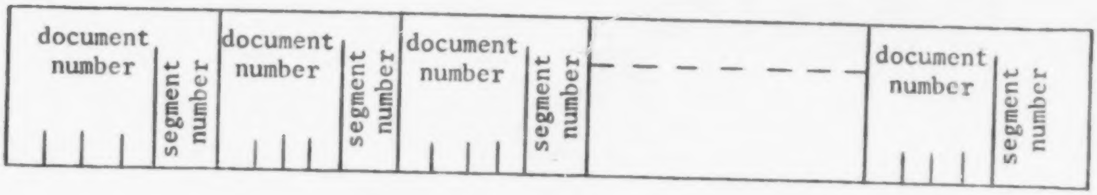
The master file record organization in this system is more complex and variable than is usual in most systems. This document describes that record organization in detail, so that the description of the programs which process it will be more easily understood.

Directories

The system maintains three directories for the purpose of retrieving documents. These directories are built as documents are stored in the device. When a document is to be stored, it is assigned a sequential internal document number. If a document is the last one in a strip, its internal number is entered in the strip directory. If it is the last one in a cell its internal number is entered in the cell directory. The track directory is written on track zero of every strip. To retrieve a document, the cell directory is searched to determine which cell it may be in. Then the strip directory is searched to determine the strip within that cell. Track 0 of this strip is then read and the track directory for this strip is then searched to determine the track. This track is read and searched for the internal number.

When a document is revised, it is tagged as deleted, given a new internal number, and written at the end of the file. These deleted documents are eliminated when the file is reorganized.

Directory Format



Continuation of Records

If the master record is longer than the capacity of a single track, it is segmented and written on more than one track. Therefore, all records contain a single byte field which serves as a continuation flag. If the record will fit in the space left on a track, the continuation flag is set to zero and the record is written in the remaining space. If it will not fit, the continuation flag is again set to zero and as many bytes as will fit are written on the track with the total length of the record in the record length field. The number of bytes written on the previous track are subtracted from the total length and the total length moved into the record length field. The continuation flag is set to one and the same procedure repeated. Thus, if the continuation flag is '1', the record is a continuation of one from a previous track.

If the record length is greater than the space left on a track, the record is continued on the next track. Using these three elements of data (the flag, the record length, and the space remaining on a track) the program is able to read back a complete logical record as required. The continuation flag and additional record length fields are eliminated. The record resides in core as logical entity with no extraneous data.

Description of Master Record

The master record is organized as follows:

1. Continuation Flag

2. Record Length

   This two byte field is the total length of the master record.

3. Internal Number

   This three byte field is an internally assigned sequential number.

4. Segment

   This one byte field is zero if this record is a searchable section. Otherwise, it represents the free text segment number (1 - 254).

5. Tag

   This one byte field represents the status of this record, that is, whether the record has been deleted.

6. Run Number

This two byte field is used for restart purposes.

7. External Number

This nine byte field contains the external number assigned to the document by the user.

8. Security

This one byte field contains security information.

9. File

This one byte field contains file identification.

10. Title

This twenty four byte field contains title information.

11. Number of Pointers

This one byte field contains the number of pointers present in this document.

12. Pointers

These seven byte fields are ordered to correspond with the alphabetic sequence of the descriptors in the document. Each one of these fields has the following subfields. There is one pointer for each descriptor in the record.

   a. Displacement

   This two byte field contains a number which, when added to the starting address of the master record, will give the address of the descriptor to which this pointer pertains.

   b. Length

   This one byte field contains the length of the descriptor.

   c. Rank

   This one byte field contains a number which indicates the relative position of each descriptor (i.e., first through nth) in the document.

d. Type

This field contains a code to indicate whether a descriptor is an expand term, main term, common descriptor, precise descriptor, temporary precise descriptor, or I.D. file descriptor.

e. Number of Categories

This one byte field contains the number of categories associated with this descriptor.

f. Number of Qualifiers

This one byte field contains the number of qualifiers associated with this descriptor.

13. Descriptors

The descriptors are stored as variable length entries, maximum length being 64 bytes. Following the descriptor there can be from 1 to 255 one byte categories associated with the descriptor. Following the categories there may be from 1 to 255 qualifiers. A qualifier is an element of information used to 'modify' or increase the specificity of a descriptor. Each qualifier is composed of two parts: a subdescriptor and up to 255 numeric values and/or links associated with the subdescriptor. The qualifier is organized as follows:

a. Length

This one byte field contains the length of the subdescriptor plus two.

b. Number of Values/Links

This one byte field contains the number of values present. If there is a link field present, it contains the number of values plus one.

c. Subdescriptor

This variable length field contains the subdescriptor. Maximum length of a subdescriptor is 64 bytes.

d. Code

This one byte field indicates the type of data that follows, i.e., whether it is a link or value. If it is a value, it also indicates whether it is a single value, lower value, upper value, time value, or date value.

e. **Value**

This five byte field contains the value converted to an internal format for compactness.

14. Note that either the subdescriptor or the values may be missing (i.e., null subdescriptor and/or null values are permissible). In the case of null subdescriptor the values that follow are associated with the descriptor.

Format examples of each record in the files are given below. The identification file record is in the same format as the master file record except that no text is allowed.

MASTER FILE RECORD

A. Searchable Section

LENGTH	CONTENTS	
1 byte	Continuation Flag	
2 bytes	Record Length	
3 bytes	Internal Number	
1 byte	Segment Number	
1 byte	Tag	
2 bytes	Run Number	
9 bytes	External Number	
1 byte	Security Code	
1 byte	File Number	
24 bytes	Title Information	
1 byte	Number of Pointers	
2 bytes	Displacement	Pointer to Descriptor Information. 255 Allowed
1 byte	Length of Descriptor	
1 byte	Rank of Descriptor	
1 byte	Type of Descriptor	
1 byte	Number of Context Codes	
1 byte	Number of Qualifiers	
1 to 64 bytes	Descriptor	
1 byte	Context Code - 255 Allowed	
1 byte	Subdescriptor Length + 2	
1 byte	Number of Values/Links	
0 to 64 bytes	Subdescriptor	
1 byte	Link/Value Code	
2 bytes	Link	
1 byte	Value Characteristic	
4 bytes	Value Mantissa	

B. Free Text Segment

LENGTH	CONTENTS
1 byte	Continuation Flag
2 bytes	Record Length
3 bytes	Internal Number
1 byte	Segment Number
1 byte	Tag
1 byte	Text Line Length + 2
1 byte	Text Line Number
1 to 71 bytes	Text Line
1 byte	Text Line Length + 2
1 byte	Text Line Number
1 to 71 bytes	Text Line

INVERTED FILE RECORD

LENGTH	CONTENTS
2 bytes	Entire Record Length
2 bytes	Binary Zeros
2 bytes	String Number
1 byte	Continuation Flag
3 bytes	Highest Document Number this String
2 bytes	Total Document Numbers this String
6 bytes	Date Last Addition
6 bytes	Date Last Deletion
1 byte	Descriptor Function
1 byte	Descriptor Field Length
$1 \leq N \leq 64$ bytes	Descriptor
3 bytes	Document Number
3 bytes	Document Number

CROSS REFERENCE RECORD

LENGTH	CONTENTS
9 bytes	External Document Number
3 bytes	Internal Document Number

MASTER FILE MAINTENANCE EDIT PROGRAM (MFEDIT)

Purpose

MFEDIT reads in the additions and/or changes to the master file and validates them for correct operations, sequences, and contents. Valid inputs are then reformatted and written on tape for use in the master file update program MFMAINT. Invalid inputs are printed out with explanatory error notes for correction of these inputs.

Input

Input to MFEDIT may be from cards or tape. Tape input must be card images, blocked by a factor of ten in the following format:

Columns 1 through 5 must contain a sequence number. Operation to be performed must start in column 6 and must immediately be followed by at least one blank. The first non-blank character following is the start of the data required for the operation. The data may be continued from one card to the next by leaving the operation field of the next card blank.

For specific types of acceptable data refer to the section on file maintenance input language.

Processing

A data set comprises all the modifications that pertain to a specific document. Each data set (i.e., all records between HDR cards) is processed completely before the output records for that data set are written on tape. Any errors encountered during the processing are printed.

A record is read and scanned for errors, both in format and content. Such things as descriptor validity and data sequence (e.g., context codes may only follow a descriptor) are checked. Each verified piece of data becomes part of a reformatted output record.

When all records of a given data set have been processed (i.e., when the next HDR record is encountered) a test is made to determine whether any errors have been found which could cause an invalid modification to the master file document. If there have been none, the output records are written on tape and the next data set is processed. If major errors are found, the entire data set is rejected. Processing terminates when the end of the input file is reached.

Output

1. Tape

   A tape of changes to the master file will be written on SYS004. This tape will provide the input to MFMAINT.

2. Listing

   A listing of all errors encountered during the edit will be printed along with diagnostics indicating the probable cause of the error and the data in which it occurred.

MASTER FILE MAINTENANCE PROGRAM (MFMAINT)

Purpose

After the change data for the master file has been edited (MFEDIT), it is submitted to MFMAINT for processing. MFMAINT updates the master file contained on the IBM data cell and creates tapes of update information for the inverted file, the cross reference file, and the I.D. file. These tapes are processed by later programs. In addition errors in data requesting modifications to the file are detected and a message indicating the nature of each is printed.

Input

The input to update the master file is arranged in MFEDIT to a format which is acceptable as input to MFMAINT. A logical input record has the following format:

BYTES	CONTENTS
0 - 1	Length of logical record
2	Operation to be performed with the following data
3 - 4	Length of data following
5 - n	Variable length data to be operated upon

Example: If it is desired to add a descriptor 20 characters long to a certain document, the input record to MFMAINT will be as follows:

BYTES	CONTENTS
0 - 1	X'0019'
2	X'01' (add descriptor)
3 - 4	X'0015'
5	Type of descriptor
6 - 25	Actual descriptor

Since most of the input records will be relatively short in length, they are blocked out of MFEDIT. MFMAINT deblocks and processes them one logical record at a time.

Processing

This program performs three basic types of operations on the master file. These are:

1. ADD (addition of a new document to the file)
2. DELETION (complete removal of a document from the file)
3. REVISION (alteration of some elements of information pertaining to some document already in the file)

The detailed specifications for preparing these inputs is given in the documentation for MDEDIT (master file edit program). The operation proceeds along the following lines. Every alteration of whatever nature is identified by a HDR (header) record, in the following pattern:

1. HDR ADD : external document number (addition)
2. HDR DEL : internal document number (deletion)
3. HDR REV : internal document number (revision)

HDR ADD records are followed by 'trailer' records, all relating to the document identified by its external number in the HDR ADD record. These trailer records contain the descriptors, subdescriptors, free text, and other elements of data pertaining to that document.

HDR DEL records may not be followed by any trailer records. A HDR DEL record will cause the indicated document to be completely purged from all the system files.

A HDR REV record is used to make changes to existing records. Like HDR ADD records, HDR REV records are followed by trailer cards. These trailer cards indicate the specific revisions to be made in the master file record.

This program gets each logical record in turn from the input tape (the tape generated by MFEDIT), determines what kind of record it is (HDR or trailer record) and performs the necessary operations. That is depending upon the nature of the input, the program will add a new document to the file, delete an existing one, or alter an existing one. The master file record, contained on the large capacity direct access file, is altered at this time. Records necessary to alter the status of the other three files (I.D. file, cross reference file, and inverted file) are also generated at this time. In addition, a tape record of the changes made, and any errors encountered, is written for later printing.

Output

A. Tape

1. Update for the inverted file will be written on SYS005. After sorting, this will provide input to IVUPDT.

2. Update to the cross reference file will be written on SYS008. After sorting, this will provide input to UPDXREF.

3. Update to the identification file will be written on SYS007. This will provide input to IDUPDT. No sorting is required.

4. Data which will be listed by MPRINT1 is written on SYS006.

B. Direct Access

The update data will be written directly into the data cell or other large capacity direct access device.

MASTER FILE PRINT PROGRAM (MPRINT1)

Purpose

MPRINT1 lists the tape of changes and errors produced by the master file maintenance program. There will be five types of records: additions (to the current master file), changes, deletions, errors (e.g., duplicate external numbers encountered), and ID. file revisions. The last type of entry will not be listed by MPRINT1. Instead it will be transferred to a tape for later processing.

Input

Input to MPRINT1 will be a tape of master file records mounted on SYS006.

Processing

MPRINT1 will process the master file entries in order by type (e.g., addition, change), listing the entire master file record acted upon. On the first pass, the addition records will be listed and the other four types written on work tapes. On the next three passes, change records, deletion records, and error records will be listed. The ID. file revision tape will be saved for later processing. Titles indicating the type of modification being listed will be printed at the top of each printer page. The output format of the records is discussed in the OUTPUT section. Processing will terminate when an end-of-file is encountered on the error tape.

Output

Four tapes will be created by MPRINT1. Three of these are work tapes and will be read back in by MPRINT1 during later processing.

1. A work tape containing change records will be written on SYS005.
2. A work tape containing delete records will be written on SYS004.
3. A work tape containing error records will be writton on SYS007.
4. A tape containing ID. file revisions will be written on SYS008. This tape, after sorting, will provide input to IDUPDT.

```
                              MASTER FILE MAINTENANCE LISTING        02/20/68          PAGE   1
                                       ADDITIONS

 INTERNAL NUMBER 1,EXTERNAL DOCUMENT NUMBER 0030010 ,FILE 1,IDENTIFICATION 00300101
   D1:GEN; S:00000000;  S:SEXF;  S:BD;  DV:06-01-66;  S:NY;  S:DR;  DV:01-10-67;  S:ND;  SV:.1F1, S:NSER;  S:REC;  S:01;
   D2:URTICARIA; S:AR;  D3:SKIN/DERM/HYSN; D4:HYSN/ATOP; D5:IMFERON; C:1; S:IRON PREPARATIONS;  S:LAKESIDE;  S:INJ;  S:CC;  SV:.2E4;
   S:IM;  S:STRT;  DV:01-10-67;  S:STOP;  DV:01-10-67;  S:DAYS;  SV:.1E1; D6:2930; D7:N10787; LK:(9); D8:HEMATOLOGIC AGENTS; C:2;
   D9:IRON PREPARATIONS; C:2

 INTERNAL NUMBER 2,EXTERNAL DOCUMENT NUMBER 0030020 ,FILE 1,IDENTIFICATION 00300201
   D1:GEN; S:00000000;  S:SEXM;  S:BD;  DV:06-29-25;  S:NY;  S:DR;  DV:01-03-67;  S:ND;  SV:.1F1; S:NSER;  S:REC;  S:HT;  SV:.65E2;
   S:WT;  SV:.3E3; S:02;  D2:URTICARIA; S:AR;  D3:SKIN/DERM/HYSN; D4:HYSN/ATOP; D5:RENOGRAFIN; C:1; S:ROENTGENOGRAPHY;  S:SQUIBB;
   S:INJ;  S:CC;  SV:.35E5; S:IV;  S:STRT;  DV:01-03-67;  S:STOP;  DV:01-03-67;  S:DAYS;  SV:.1F1; D6:X931; D7:N10040; LK:(9);
   D8:IODINATED COMPDS; C:4; D9:ROENTGENOGRAPHY; C:2

 INTERNAL NUMBER 3,EXTERNAL DOCUMENT NUMBER 0030030 ,FILE 1,IDENTIFICATION 00300301
   D1:GEN; S:00611558;  S:SEXF;  S:BD;  DV:01-16-67;  S:NY;  S:DR;  DV:01-16-67;  S:ND;  SV:.1E1; S:NSER;  S:REC;  S:HT;  SV:.66E2;
   S:WT;  SV:.13E3; S:01;  D2:TREMOR; S:AR;  D3:NER/CNS/CRN; D4:RENOGRAFIN; C:1; S:ROENTGENOGRAPHY;  S:SQUIBB;  S:INJ;  S:CC;
   SV:.2E6; S:IV;  S:STRT;  DV:01-16-67;  S:STOP;  DV:01-16-67;  S:DAYS;  SV:.1E1; D5:X931; D6:N10040; LK:(9); D7:IODINATED COMPDS;
   C:4; D8:ROENTGENOGRAPHY; C:2

 INTERNAL NUMBER 4,EXTERNAL DOCUMENT NUMBER 0040020 ,FILE 1,IDENTIFICATION 00400201
   D1:GEN; S:00118155;  S:SEXM;  S:BD;  DV:01-14-22;  S:ARK;  S:DR;  DV:01-09-67;  S:ND;  SV:.3F1; S:NSER;  S:REC;  S:HT;  SV:.68E2;
   S:WT;  SV:.15E3; S:01;  D2:RASH MAC PAP; S:AR;  D3:SKIN/DERM/ERY; D4:HYSN/NEC; D5:PHENOBARBITAL; C:1; S:SEDATIVES-HYPNOTICS;
   D6:IAPOSS; S:TAB;  S:MG;  SV:.18F6; SV:.12F4; S:ORAL;  S:STRT;  DV:01-08-67;  DV:01-06-67;  S:STOP;  DV:01-09-67;  DV:01-06-67;
   S:DAYS;  SV:.2E1; SV:.1E1; D7:3240; D8:MORPHINE SULFATE; C:1; S:NARCOTIC ANALGESICS;  D9:7817; D10:BARBITURATES; C:4;
   D11:NERVOUS SYSTEM DRUGS; C:2; D12:SEDATIVES-HYPNOTICS; C:2; D13:NARCOTIC ANALGESICS; C:2; D14:OPIUM ALKALOIDS; C:4

 INTERNAL NUMBER 5,EXTERNAL DOCUMENT NUMBER 0080020 ,FILE 1,IDENTIFICATION 00800201
   D1:GEN; S:63024046;  S:SEXF;  S:BD;  DV:00-00-11;  S:NY;  S:DR;  DV:01-19-67;  S:ND;  SV:.1F1; S:NSER;  S:REC;  S:01;  D2:DYSPNEA
   S:AR;  D3:RES/FUNC; D4:NICOTINIC ACID; C:1; S:VITAMIN B COMPLEX;  S:TAB;  S:MG;  SV:.5E5; S:ORAL;  S:STRT;  DV:01-19-67;  S:STOP;
   DV:01-19-67;  S:DAYS;  SV:.1F1; D5:3689; D6:VASODILATORS; C:2; D7:VITAMIN B COMPLEX; C:2

 INTERNAL NUMBER 6,EXTERNAL DOCUMENT NUMBER 0080020 ,FILE 1,IDENTIFICATION 00800202
   D1:GEN; S:63024046;  S:SEXF;  S:BD;  DV:00-00-11;  S:NY;  S:DR;  DV:01-19-67;  S:ND;  SV:.1E1; S:NSER;  S:REC;  S:01;
   D2:RASH MAC; S:AR;  D3:SKIN/DERM/CUT; D4:HYSN/NEC; D5:NICOTINIC ACID; C:1; S:VITAMIN B COMPLEX;  S:TAB;  S:MG;  SV:.5E5; S:ORAL;
   S:STRT;  DV:01-19-67;  S:STOP;  DV:01-19-67;  S:DAYS;  SV:.1F1; D6:3689; D7:VASODILATORS; C:2; D8:VITAMIN B COMPLEX; C:2

 INTERNAL NUMBER 7,EXTERNAL DOCUMENT NUMBER 0080030 ,FILE 1,IDENTIFICATION 00800301
   D1:GEN; S:00005914;  S:SEXM;  S:BD;  DV:08-15-15;  S:NY;  S:DR;  DV:02-00-65;  S:ND;  SV:.2E1; S:SFR;  S:A/S;  S:WT;  SV:.14E3;
   S:02;  D2:LYMPHADENT; S:AR;  D3:HAL/LYM/GLN; D4:HYSN/SST; D5:DILANTIN; C:1; S:ANTICONVULSANTS;  S:PARKE DAVIS;  D6:IAREM; S:CAP;
   S:MG;  SV:.3E6; SV:.9E5; S:ORAL;  S:STRT;  DV:00-00-61;  S:STOP;  DV:00-00-67;  S:DAYS;  SV:.219E4; S:TAB;  D7:3531; D8:N16480;
   LK:(9); D9:PHENOBARBITAL; C:1; S:SEDATIVES-HYPNOTICS;  D10:ANTICONVULSANTS; C:2; D11:HYDANTOINS; C:4; D12:BARBITURATES; C:4;
   D13:NERVOUS SYSTEM DRUGS; C:2; D14:SEDATIVES-HYPNOTICS; C:2

 INTERNAL NUMBER 8,EXTERNAL DOCUMENT NUMBER 0090090 ,FILE 1,IDENTIFICATION 00900901
   D1:GEN; S:00019061;  S:SEXM;  S:BD;  DV:00-00-18;  S:NY;  S:DR;  DV:11-06-66;  S:ND;  SV:.2E1; S:NSER;  S:REC;  S:HT;  SV:.67E2;
   S:WT;  SV:.148E3; S:01;  D2:HEMATURIA; S:AR;  D3:GU/UT/URN; D4:HMRG; D5:COUMADIN; C:1; S:ANTICOAGULANTS;  S:ENDO;  S:TAB;  S:MG;
   SV:.75E4; S:ORAL;  S:STRT;  DV:10-08-66;  S:STOP;  DV:11-05-65;  S:DAYS;  SV:.6E2; D6:3320; D7:N09218; LK:(9); D8:ANTICOAGULANTS;
   C:2

 INTERNAL NUMBER 9,EXTERNAL DOCUMENT NUMBER 0130030 ,FILE 1,IDENTIFICATION 01300301
   D1:GEN; S:00000000;  S:SEXF;  S:BD;  DV:05-06-21;  S:TEX;  S:DR;  DV:01-19-67;  S:ND;  SV:.1E1; S:NSER;  S:REC;  S:WT;  SV:.154E3
   S:02;  D2:RASH; S:AR;  D3:SKIN/DERM/ERY; D4:HYSN/NEC; D5:DILANTIN; C:1; S:ANTICONVULSANTS;  S:PARKE DAVIS;  S:CAP;  S:MG;
   SV:.3E6; S:ORAL;  S:STRT;  DV:01-19-67;  S:STOP;  DV:01-23-67;  S:DAYS;  SV:.5E1; D6:4469; D7:0; LK:(9); D8:N16480; LK:(9);
   D9:ANTICONVULSANTS; C:2; D10:HYDANTOINS; C:4

 INTERNAL NUMBER 10,EXTERNAL DOCUMENT NUMBER 0130040 ,FILE 1,IDENTIFICATION 01300401
```

ID. FILE UPDATE PROGRAM (IDUPDT)

Purpose

As a result of a master file update run, certain ID. file descriptors may be added, changed, or deleted from the master file. This program updates the ID. file so it will correspond with the status of these descriptors in the master file.

Input

Three tapes will be acceptable as input to IDUPDT. They are:

1. The tape containing the prior ID. file will be mounted on SYS005.
2. The tape containing new additions to the ID. file will be mounted on SYS007.
3. A third tape which contains revisions to already existing ID. file records will be mounted on SYS004.

Processing

IDUPDT reads records from the revisions and merges them with the records from the prior ID. file. When the document numbers are equal and a delete tag is set in the revision record, the prior ID. file record thus identified is written onto the print tape. This can be printed using MPRINT2 for a listing of deleted ID. file records. Otherwise, the input record is written on the output tape. When an end-of-file is read from the revisions tape, records are read from the additions tape and placed at the end of the new ID. file tape until end-of-file is reached on the additions tape which terminates the program.

Output

1. The new ID. file tape will be written on SYS001.

2. Any deleted records will be written on a tape mounted on SYS006. This tape may be listed by MPRINT2, a slightly modified version of MPRINT1.

UPDATE CROSS REFERENCE FILE (UPDXREF)

Purpose

UPDXREF updates the cross reference file for the master file. The cross reference file is the ordered listing of the external/internal document numbers resident on disk. The pairs are in order according to the collating sequence of the external number. The master file maintenance program uses the cross reference file as a validity check on modification requests to the master file. For example, a check can be made to determine whether or not a document which is to be added is, in fact, already in the file.

Input

UPDXREF uses two tapes and a disk as input. They are:

1. The sorted tape created by MFMAINT consisting of update to the cross reference file will be mounted on SYS004.
2. The tape containing the prior cross reference file will Be mounted on SYS007.
3. The cross reference disk is mounted on SYS018.

Processing

The two tapes are read and the records are merged in order by external document numbers. Since the tape created by the master file maintenance program will contain both additions and deletions, these records are tested for type and the indicated functions performed (i.e., if a record specifies delete, the record referenced is located and deleted). The tape containing the prior cross reference file is read and acted upon. Processing continues until an end-of-file is encountered on both tapes.

Output

UPDXREF creates two types of output. They are:

1. Tape

   The tape containing the new cross reference file will be written on SYS008. This tape may be listed by XREFPRNT.

2. Disk

   The direct access version of the cross reference file will be written on SYS018.

CROSS REFERENCE PRINT PROGRAM (XREFPRNT)

Purpose

This program provides a listing, for maintenance purposes, of the cross reference file.

Input

The cross reference tape file is mounted on SYS004.

Processing

XREFPRNT lists external/internal numbers in eight columns. The external number is compared to the previous external number processed, since duplicate external numbers may occur if the indexer has overridden the duplication option in MFMAINT. All internal numbers are listed below the external number to which they are related. Processing terminates when the end-of-file is encountered on the input tape.

Output

A printed listing of the cross reference file is output. A sample of this listing is shown below.

```
                                        CROSS-REFERENCE LISTING                                      11/01/67

  EXTERNAL      EXTERNAL      EXTERNAL      EXTERNAL      EXTERNAL      EXTERNAL      EXTERNAL      EXTERNAL
     INTERNAL      INTERNAL      INTERNAL      INTERNAL      INTERNAL      INTERNAL      INTERNAL      INTERNAL

     1             29            399           4999          59999         699999
           1             31            61           2641         10321         18001
           2             32            62           2897         10577         18257
           3             33            63           3153         10833         18513

     2             39            499           5999          69999         799999
           4             34            64           3409         11089         18769
           5             35            65           3665         11345         19025
           6             36            66           3921         11601         19281

     3             49            599           6999          79999         899999
           7             37            67           4177         11857         19537
           8             38            68           4433         12113         19793
           9             39            69           4689         12369         20049

     4             59            699           7999          89999         999999
          10             40            70           4945         12625         20305
          11             41            71           5201         12881         20561
          12             42            72           5457         13137         20817

     5             69            799           8999          99999
          13             43            73           5713         13393
          14             44            74           5969         13649
          15             45            75           6225         13905

     6             79            899           9999          199999
          16             46            76           6481         14161
          17             47            77           6737         14417
          18             48            78           6993         14673

     7             89            999           19999         299999
          19             49            79           7249         14929
          20             50            80           7505         15185
          21             51            81           7761         15441

     8             99            1999          29999         399999
          22             52           337           8017         15697
          23             53           593           8273         15953
          24             54           849           8529         16209

     9             199           2999          39999         499999
          25             55          1105           8785         16465
          26             56          1361           9041         16721
          27             57          1617           9297         16977

    19             299           3999          49999         599999
          28             58          1873           9553         17233
          29             59          2129           9809         17489
          30             60          2385          10065         17745
```

D52

INVERTED FILE UPDATE PROGRAM (IVUPDT)

Purpose

Whenever the master file is updated by MFMAINT, a tape of the modifications which influence the inverted file is created. This tape is used by the inverted file update program to alter the inverted file tape and the inverted file disk. In this manner, the inverted file will always reflect the current status of the master file.

Input

Input consists of two tapes: the sorted inverted file update and the prior inverted file.

1. Inverted file update tape is mounted on SYS004.

2. Prior inverted file tape file is mounted on SYS006.

Processing

The record changes (inverted file update from the master file update program) will be organized alphabetically by main terms, and, if more than one change is requested per main term record, by document number. The two tapes (prior inverted file and sorted update) will be processed concurrently and comparisons made between the main terms of each record on either tape to determine whether or not the prior inverted file record being scanned is to be altered. These comparisons will have the following significance:

1. If the update main term is low, an inverted file record will be created for this term and update records immediately following which refer to this new record will be processed. The entire new record thus created will be output.

2. If the update main term is high, it refers to a later record on the prior inverted file tape. Then the prior inverted file record will be output unchanged.

3. If the update main term is equal to the prior inverted file main term, the type of alteration is checked. If it is a deletion of the entire record, a message indicating the nature of the change is written on SYSLST and the next record from both input tapes is read. If it is not a 'delete all' request, the change requested is made, all indicated modifications are made, and the next update record is read. When all records referring to this record have been processed, the altered record is output.

As soon as an end-of-file condition is encountered on either tape, the main term area of that tape file is filled with high values and the file is closed. In this manner, any compare will indicate that the other tape record is low and is to be processed first. When an end-of-file condition occurs on both tapes, the file is closed and processing terminates.

Output

Output consists of one tape file, one disk file, and a listing of any errors encountered during the update run.

1. Tape

   The updated inverted file tape will be written on SYS008. This tape record will contain statistical information regarding the file as well as the terms and their document numbers. In addition, all descriptors which are input to the inverted file will reside on this tape.

2. Disk

   The inverted file itself will reside on a disk. It will be composed only of precise and temporary descriptors and their associated document numbers. The disk will be created on SYS014.

INVERTED FILE PRINT PROGRAM (IFPRNT)

Purpose

For maintenance purposes it will occasionally be necessary to obtain a complete listing of all of the members of the inverted file. This program furnishes a formatted dump of the entire file on tape.

Input

The inverted file tape will be mounted on SYS009.

Processing

IFPRNT reads the inverted file tape and prints each precise or temporary descriptor with its associated internal document numbers. When a common descriptor is encountered, a message is printed below the term indicating the total number of documents in which it occurs. Special descriptors (unverified descriptors) are printed at the end of the listing and are preceded by a dollar sign ($). The document numbers with which these special descriptors are associated are also listed. When the end of the inverted tape file is reached, processing terminates.

Output

The file will be listed on the printer. A sample of the output listing is shown below.

ACETYLDIGITOXIN
22, 31

ACETYLSALICYLIC ACID
14, 15, 17, 42, 72, 76, 88, 107, 110, 111, 116, 159

ADRENAL CORTICOSTEROID
75

ADRENERGIC AGENTS
1, 20, 110

AGITATION
42, 95, 96

ALCOHOL
129

AMBLYOPIA
82

AMINOPHYLLINE
2, 143

AMINOSALICYLATE SODIUM
127

AMINOSALICYLIC ACID
78, 80, 109, 111, 128, 170

AMOBARBITAL
90

AMOBARBITAL SODIUM
10

AMPICILLIN
16, 20, 25, 34, 54, 105, 106, 107, 110, 155, 168

AMPICILLIN TRIHYDRATE
139

AMYTAL
90

ANALGESIC-ANTIPYRETIC
14, 15, 17, 42, 72, 76, 88, 107, 110, 111, 116, 129, 159

ANAPHYL/CIRCULAT/FAIL
7, 167

INVERTED FILE STATISTICS PRINT PROGRAM (IVPRINT)

Purpose

In order to gain a rapid status of the descriptors used to index documents, an indication of the activity of each of these may be desirable. The statistics print program will provide data such as when was the descriptor last used and how many documents does it describe. This information may aid the indexer in the use of the descriptors. For example, if a descriptor is used in a great many documents, it may not have the precision required for rapid retrieval. In this case, it may be necessary to change its function to common instead of precise.

Input

The inverted file tape will be mounted on SYS007. A control card which has the desired header will be read on SYSRDR. All data on the card will be printed.

Processing

IVPRINT processes the statistical data preceding the document strings on the inverted file tape. The term is printed along with the date it was last added to the master file as a descriptor, the date it was last deleted from the master file, the total number of documents in which it is used, and the function of the descriptor. Each term is processed separately. Processing terminates when an end-of-file is encountered on the input tape.

Output

The printed statistics will be written on SYSLST. The special FDA carriage tape should be used to ensure correct processing. This carriage tape contains only a channel 1 punch. An example of the type of output expected is shown below.

```
            INVERTED FILE STATISTICS PRINT PROGRAM                                              PAGE  1
DESCRIPTOR                                        COUNT     LAST ADDITION  LAST DELETION  FUNCTION
ACETYLDIGITOXIN                                       2     06/11/68                      PRECISE
ACETYLSALICYLIC ACID                                 12     06/11/68                      PRECISE
ADRENAL CORTICOSTEROID                                1     06/11/68                      PRECISE
ADRENERGIC AGENTS                                     3     06/11/68                      PRECISE
AGITATION                                             3     06/11/68                      PRECISE
ALCOHOL                                               1     06/11/68                      PRECISE
AMBLYOPIA                                             1     06/11/68                      PRECISE
AMINOPHYLLINE                                         2     06/11/68                      PRECISE
AMINOSALICYLATE SODIUM                                1     06/11/68                      PRECISE
AMINOSALICYLIC ACID                                   6     06/11/68                      PRECISE
AMOBARBITAL                                           1     06/11/68                      PRECISE
AMOBARBITAL SODIUM                                    1     06/11/68                      PRECISE
AMPICILLIN                                           11     06/11/68                      PRECISE
AMPICILLIN TRIHYDRATE                                 1     06/11/68                      PRECISE
AMYTAL                                                1     06/11/68                      PRECISE
ANALGESIC-ANTIPYRETIC                                13     06/11/68                      PRECISE
ANAPHYL/CIRCULAT/FAIL                                 2     06/11/68                      PRECISE
ANEMIA                                                2     06/11/68                      PRECISE
ANEMIA HEMOLYTIC G6PD                                 1     06/11/68                      PRECISE
ANGIOEDEMA                                            2     06/11/68                      PRECISE
ANTI-INFECTIVES                                      52     06/11/68                      PRECISE
ANTI-INFLAMMATORY AGENT                               2     06/11/68                      PRECISE
ANTIACID-FAST DRUGS                                  14     06/11/68                      PRECISE
ANTIARTHRITIC-RHEUMATIC                               1     06/11/68                      PRECISE
ANTIBACTERIALS TOPICAL                                2     06/11/68                      PRECISE
ANTIBIOTICS                                          55     06/11/68                      PRECISE
ANTICOAGULANTS                                        3     06/11/68                      PRECISE
ANTICONVULSANTS                                       4     06/11/68                      PRECISE
```

DATA CELL DUMP PROGRAM

BLOCK CHANGE EDIT PHASE (EDITBC)

Purpose

When a term is to be replaced or deleted throughout the entire master file, the modification request is submitted to this phase. This phase will test the validity of the requests and build a table of these modifications. Then, during reorganization of the data cell, that table is used to update the master file.

Input

EDITBC reads cards in the following format:

```
01BCDELETE PHRENOLOGY

02BCREPLACE ACUTE POLIOMYELITIS:POLIOMYELITIS
```

Only a 'replace' may be continued on the next card. The format of the card is checked and a message printed if the format is in error. However, only the term which will be used as a replacement is checked for validity (i.e., is compared to a valid term on the dictionary master file). If a term to be altered is misspelled, it will either not be found or the wrong term may be altered. Extreme care should be exercised when using this option.

Processing

This phase reads an input card and checks the format for errors. If none are found, the card type (replace or delete) is determined. If the card is a delete, the term is moved to the output area, the delete code is moved in, and the next card is read. In the case of a replace, the term and its replacement are moved to the output area (with the length of each) and the replace code is moved in. Since a replace card may be continued, the next card is tested. If it is a continuation card, the remainder of data is moved to the output area. If it is not a continuation card, it is processed as above.

When an end-of-file occurs on the input device, the data in the output area is sorted and passed on to the next phase.

DATA CELL DUMP PROGRAM

DUMP PHASE (DCELLDMP)

Purpose

The purpose of this phase is to dump the master file onto magnetic tape.

Input

Input to DCELLDMP will be data cell records. The data cell will be assigned to SYS031.

Processing

This phase reads records from the data cell, performs block changes, and writes out a blocked master file tape. Tapes of update information for the inverted file, cross reference file, and ID. file are also created as a result of block changes.

Output

1. Output will be a magentic tape file containing the entire master file. The tape will be assigned to SYS007.
2. Unsorted inverted file update will be assigned to SYS004.
3. Unsorted cross reference update will be assigned to SYS005.
4. ID. file revisions will be assigned to SYS006.

DATA CELL RELOAD PROGRAM (LOADCELL)

Purpose

When the tape of the updated master file has been created, the file is reloaded onto the data cell by LOADCELL.

Input

The tape created by DCELLCMP is used as input to LOADCELL. It will be mounted on SYS004.

Processing

LOADCELL simply reads the tape records and writes them onto the data cell, creating and updating an index to the master file during the processing. The records are transferred to MPRINT1 which acts as a subroutine to LOADCELL. Processing terminates when the end-of-file is encountered.

Output

The records are written by LOADCELL onto the data cell. In addition, there is an option to list the entire master file. MPRINT1 is used for this purpose. The printer format is described in the MPRINT1 documentation.

THE SEARCH SYSTEM

SEARCHING SYSTEM

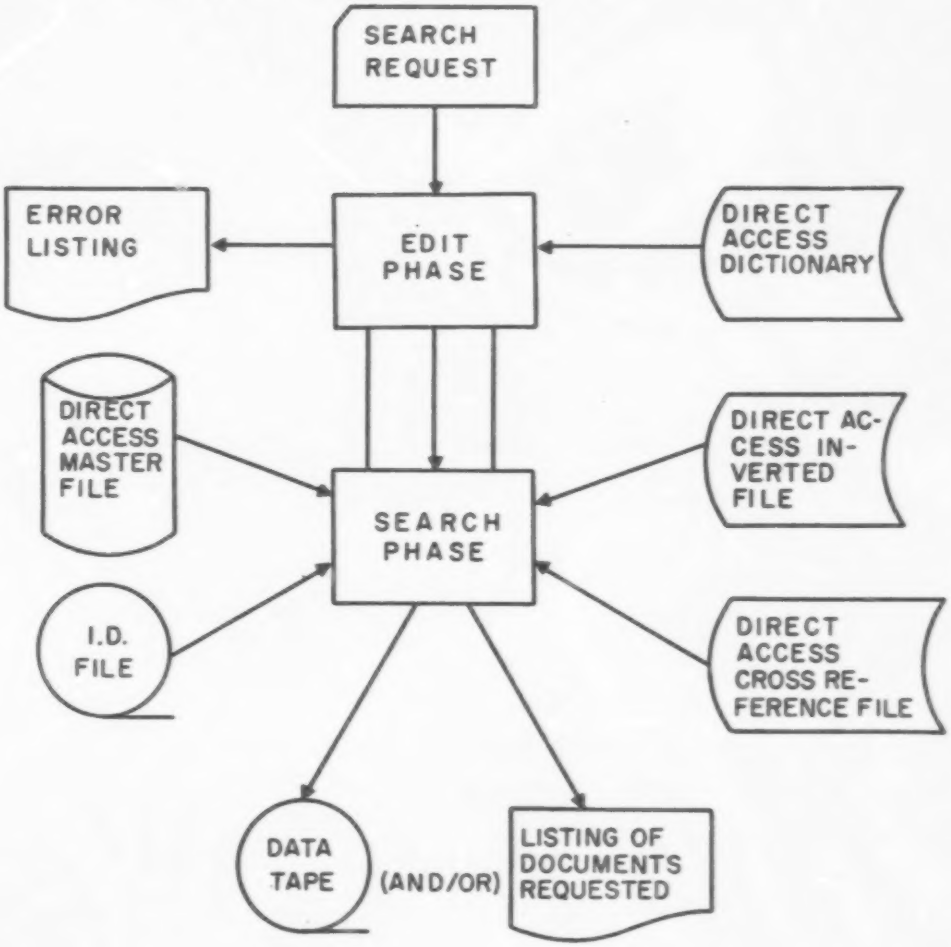
The data files described in the preceding sections are maintained so that information pertaining to a given topic may be extracted as needed, in the form desired. The searching system is designed to perform this extraction process.

Unlike the dictionary and master file maintenance systems, the search system consists of a single program, not a series of programs. Given a request for information from the files, the search program will first validate that request to ensure that the request terms are, in fact, valid ones. This is done by checking each term against the same dictionary (on disk files) used to validate the terms used in the documents themselves.

If the request contains no errors, the actual searching process begins. The descriptors (terms) of the request are first looked up in the inverted file, a process directly analogous to the searching of a printed index. From the inverted file, the program constructs a list of document numbers identifying those documents which are potential responses to the request. This list is termed the 'candidate list'. The program then inspects in detail every document identified on this candidate list to determine whether or not the document does, in fact, match the exact specifications of the request. If it does, that document record is written out. If not, the program examines the next member of the list. It proceeds in this manner until the candidate list is exhausted. At this point, the documents which satisfy the request (the 'hits) have been printed, or are recorded on a tape, or both, depending upon the options selected by the requestor. The records on tape are suitable for further processing by special purpose programs, such as a report generator program to produce statistical summaries or other forms of reports.

As is evident from the description above, the program processes each request an an entity, completing all action on one request before beginning work on another. Thus, while it is possible to 'batch' requests by stacking them one behind the other in the card reader, the program treats each as a 'batch of one'. This design method was adopted to permit modification of the manner in which requests are submitted to accomodate remote terminals at some future time, if desired.

SEARCH SYSTEM


SEARCH REQUEST
ERROR LISTING
EDIT PHASE
DIRECT ACCESS DICTIONARY
DIRECT ACCESS MASTER FILE
DIRECT ACCESS INVERTED FILE
SEARCH PHASE
I.D. FILE
DIRECT ACCESS CROSS REFERENCE FILE
DATA TAPE
(AND/OR)
LISTING OF DOCUMENTS REQUESTED

SEARCH PROGRAM INPUT SPECIFICATIONS

Introduction

This section describes the preparation of inquiries, or requests, for the data files. It includes a description of the card formats used, the options available, and the type of output available.

Input Formats

A single inquiry or request to the system is prepared in punched card form. The general pattern of the cards is shown below. Columns 1 and 2 are reserved for identification of the particular request. Columns 3 and 4 provide a sequence number for cards within a request. Column 5 is the beginning of the word which identifies the type of card; a colon is used to mark the end of the identifying word. The remainder of the card is punched with appropriate data as described below. There are six different types, some required and some optional.

1. XXNNTITLE:
2. XXNNSEARCH:
3. XXNNPRINT:
4. XXNNREPORT:
5. XXNNSELECT:
6. XXNNREQUEST:

Description of Card Types

1. Title Card(s)

   The requestor may enter any kind of identifying data he wishes: name, address, telephone number, name of the search, or any other information. The data may extend to column 80 of the first card, and onto a second card, beginning at column 12 or later, if desired. There must be a minimum of one, and a maximum of two title cards. An example of a pair of title cards is shown below.

```
0101TITLE:  J.F. JACKSON, PHARMACOLOGY DEPT. RM 324, EXT 7741

0102TITLE:      SEARCH ON CHLOROPROMAZINE DERIVATIVES.
```

2. Search Card

This card is used to supply various overall parameters to the searching program. It contains information specifying which TYPE of search, which FILE(s) are to be searched, which DATA BASE(s) are to be searched, and the MAXIMUM number of responses desired. Each of these various parameters is described below.

a. TYPE of Search

There are three possible kinds of searches: document, Boolean, and mixed. A document (document number) search is one in which only certain specified documents, as identified by their internal document numbers, are retrieved. That is, given a list of document numbers, the system will retrieve those records and print them out. A Boolean search is one in which documents are retrieved if they contain some specified combination of descriptors, subdescriptors, and the like. A mixed search is a combination of the above two (i.e., certain specified documents, as identified by their document numbers, will be retrieved if and only if the records for those documents contain the specified descriptors and/or subdescriptors).

b. File to be Searched

The word 'file' in that instance refers to the physical computer files on which the data is kept. There are three such files: the master file, which contains the full record for every document in the system including free text, the inverted file, which is an index to the master file, and the ID. file, which is a condensed version of the master file. The ID. file contains only some of the descriptors relating to each document, and contains no free text. The choice of which file to search depends upon the aims of the searcher. The ID. file search can be used whenever the searcher wants to get an overview of the file contents using very broad (common) descriptors. The inverted file search could be used as a 'first pass' at some topic to gain some idea of how much data there is in the file on that topic.

Of these three files, the amount of information produced at the end of the search and the amount of time required for the search vary proportionately. This is illustrated below.

KIND OF SEARCH	AMOUNT OF INFORMATION	SPEED
Master File	Maximum: all searchable data, all free text, as desired	Slowest
ID. File	Title, document number part of searchable section, no text	Moderately Fast
Inverted File	Document number only	Very Fast

c. Data Base(s) Searched

As mentioned above, the word 'file' in this search card refers to the physical organization of the data. Each logically distinct group of information may be identified as a separate data base (e.g., 'Human Adverse Reactions', 'Veterinary data', 'Legal Hearings', etc.). The user may specify that either all, or only certain specified ones of these data bases be searched.

d. Maximum Number of Responses

The system contains a built-in upper limit on the number of responses to a given question. However, the requestor can override this limit by specifying a number of his own choice. The system will accept this specified number in lieu of the internal limit. The specified number may be either larger or smaller than the system limit.

All of the options in the search card are shown below. For each of the four parameters there is a preferred (automatic) option which the system will use if nothing is specified. That preferred option is underlined below.

```
                BOOLEAN        ID FILE                  ALL
SEARCH : TYPE : MIXED; FILE :  MASTER; DATA BASE :  HUM; MAX : system limit
                DOCUMENT       INVERTED                 VET*        user  number
```

It is therefore acceptable to use a search card with nothing punched in it beyond the word SEARCH :. Such a card would automatically result in a Boolean search of the master file, searching all data bases and producing up to the system limit of responses.

* Multiple data bases may be specified (e.g., DATA BASE : HUM, VET, LGL; is permitted

3. This card informs the system of how much data the requestor wishes to have printed out once a document which satisfies the requirements of the search logic has been found. There are two print parameters, one controlling the printing of the searchable section and the other controlling the printing of the free text segments. The searchable section contains the actual terms used for search purposes, while the free text, which can extend over a number of segments or paragraphs, carries only descriptive information not used in the search process itself. The requestor may choose to print, or not, the terms in the searchable section, and may choose to have all, none, or selected segments of the free text printed out. This is illustrated below.

```
                          ALL
PRINT : TERMS : NO; TEXT : NONE
                YES        S1,S2,S3,...(up to a maximum of ten)
```

Again, the underlining indicates the preferred option, which will automatically be chosen by the program if nothing is specified. Note that if the requestor does not specify any options, thus permitting the program to choose NO and NONE, the only result of the search, even though the master file is searched, will be the list of document numbers satisfying the request. It should also be noted that it is illogical, and therefore erroneous, to specify any text with an inverted file search, since there is no text available in that file. It is also invalid to specify available for printing from that file.

4. Report Card

This card is used only if the requestor whishes to extract data from the files for further computer processing. This is known as the 'RPG Option', which permits the user to not only have the results of his searches printed for visual inspection, but also to have the extracted data written onto magnetic tape for use in preparing special reports of various kinds. If the user does not wish to prepare these reports, the report card and the select card described below may be omitted.

The report card specifies the general nature of the information to be selected for the report tape. There are two parameters to be given. The first is REQID (requestor identification). This consists of any four character set of symbols the user desires. Its function is to uniquely identify this particular group of selected information so that it may be identified by the programs which are to follow. The second parameter is OUTPUT, which identifies the kind of information which the user wants selected. There are four possibilities: descriptors, subdescriptors, values, and text. An example of the report card follows:

REPORT : REQID : 1234; OUTPUT : DESCR, SUBDESCR, TEXT;

This card indicates to the program that, from the records which satisfy the requirements of the search request, the user desires a tape file which contains records for every descriptor, for every subdescriptor, and for all free text segments of those document records.

5. Select Card

This card is used to further specify the nature of the data desired for the RPG Option. It permits the user to select descriptors, subdescriptors, and values associated with only certain file/context codes. It also allows him to specify which free text segments he wants. An example is given below.

SELECT : DESCR : F/C, F/C, F/C,...TEXT : S1,S2,...
(F/C represents desired file/context codes) (maximum of ten text segments)

This select card will result in the extraction of only those descriptors, subdescriptors, or values which are associated with the specified file/context codes. Similarly, only the specified text segments will be extracted. The entire record will be selected by specifying ALL following both DESCR : and TEXT : above.

6. Request Card

This card, or group of cards, is used to hold the actual search terms for the Boolean, document, or mixed search. The detailed explanation of these request cards is given in the 'Request Language Description'.

SEARCH LANGUAGE DESCRIPTION

Since the PL/1 character set is the basic standard in this system, the following are acceptable in descriptors and subdescriptors:

A through Z; 0 through 9;) (. , * / - % # @ ' " _ = + therefore, the following characters are used in the search language as separators and operators:

1.	&	logical AND
2.	¬	logical NOT
3.	|	logical OR
4.	:	colon (for numeric operator, EQ:)
5.	;	semi-colon (used as a comma)
6.	$	for truncation of descriptors
7.	>	right parentheses
8.	<	left parentheses
9.	?	end of request

The request language (the means of interrogating the file of indexed documents) is designed to provide a wide range of capabilities, from simple 'AND', 'OR', 'NOT' statements to complex statements involving sequences and linkage of descriptors.

The fundamental abilities of the language permit the usual three Boolean operators, 'AND', 'OR', and 'NOT'. The symbols '&', '|', and '¬' respectively, will be used to represent these operators. A descriptor, as interpreted by the search language, is that which appears between two operators. That is, a descriptor, with its associated subdescriptors and values is treated as a single complex descriptor. In the following example,

A&B&C,

can represent a simple inquiry such as,

```
 (A)     (B)      (C)
GANATOL & RASH & RECOVERED
```

or can represent a more complex statement, such as,

```
                    (A)                              (B)             (C)
GANATOL; RTE ADM ORAL; DAY DOSE GM; EQ : 2.0 & RASH; SEVERE & RECOVERED
```

In the first instance, the three elements of the questions separated by 'AND' operators are simple descriptors. In the second instance, descriptor A is a complex one represented by the entire statement 'Ganatol; Rte Adm Oral; Day Dose Gm; EQ : 2.0', which is itself composed of:

A descriptor	Ganatol
A subdescriptor	Rte Adm Oral
A second subdescriptor	Day Dose Gm
A single value	2.0

Descriptor B is represented by 'Rash; Severe', which is composed of:

A descriptor	Rash
A subdescriptor	Severe

Descriptor C is represented by the simple descriptor 'Recovered', as in the first example.

Unless specifically stated otherwise, the term 'descriptor' will be taken to mean either a simple or a complex descriptor in the remainder of this document. Rules for the formulation of complex descriptors are presented later.

Descriptors may be combined in a large number of ways. The simplest of these ways, a simple 'ANDing', has already been illustrated, as A&B&C. The 'OR' is used in a similar manner, A|B|C. The former states that all three descriptors, A, B, and C, must appear on a document in order for that document to satisfy the request (be a 'hit'). The latter states that if any one of the three descriptors appears in a document, the document is hit.

Parentheses (less than and greater than signs) may be used freely to alter the meaning, but must be used with care. Parentheses cause the entire expression enclosed within them to be evaluated as an entity, the result of that evaluation then being considered as a sort of 'super-descriptor', which is then analyzed with the remainder of the request. For example, consider the following:

1. A|B&C
2. <A|B>&C
3. A|<B&C>

The first statement defines a hit as any document which contains either A or both B and C.

The second defines a hit as one which contains either A or B and C. The third defines a hit as one which contains either A or both B and C, which is the same as (1). This is an example of redundant parentheses, which will not affect the search logic.

Nested parentheses are permissible, as in the following:

A&<B&<C|D>|E>

This decomposes into A&B&D or A&E. However, if the parentheses are modified the same expression decomposes to different meanings.

Removing the outer parentheses gives A&B&<C|D>|E , which decomposes into A&B&C or A&B&D or E.

Removing the inner parentheses instead gives A&<B&C|D|E> , which decomposes into A&B&C or A&D or A&E.

Removing both sets gives A&B&C|D|E , which decomposes into A&B&C or D or E.

Logically, there is no limit to the number of nested parentheses which are possible, since the logical system deals with them one at a time. However, other limitations impose a maximum of eight nested sets. This does not imply that only eight sets may be used in an inquiry, only that a maximum of eight nested sets are permitted.

The 'NOT' operator (¬) should be treated with special care since its function is different from either & or |. AND (&) and OR (|) are binary operators which imply a relationship between two descriptors. 'NOT' is unary. It affects only the descriptor following the ¬. 'NOT' implies 'AND', i.e., A¬B = A and not B. A&(B|¬C) is not permitted and must be written as A&B|A¬C.

Two special capabilities of logical manipulation of descriptors are allowed, in addition to the elementary Boolean operators. These are: linkage and sequence.

1. Linkage

Links perform a special qualifying function for descriptors and subdescriptors. To specify that descriptors are to be linked in a search request, the descriptors in question are to be enclosed in parentheses preceded by the letter 'L' as below:

The 'AND' operator (&) must appear between descriptors. Neither logical 'OR' (|) nor logical 'NOT' (¬) is valid. Furthermore, no further parentheses may appear within the parentheses enclosing the linked descriptors (i.e., no nesting). If links are used, the document will be a hit only if the descriptors appear in the record and if they share at least one link in common.

Also partial linking of descriptors can be requested such that the first two descriptors must share at least one link in common, the third descriptor must be linked with the first or second descriptor or both, the fourth with the first, second or third, and so on. To request this type of linkage, precede the descriptors within parentheses with 'L2'. This is called 'ACCUMULATIVE' linking.

There are three additional linking options which involve linking of subdescriptors. The 'L12' option combines the links of those subdescriptors specified in the input with their associated descriptors before testing if the descriptors are linked together. So if 'L12' is requested, the document will be a hit only if the descriptors and subdescriptors appear in the record and if each descriptor/subdescriptor set is linked with each other such set within the parentheses. The option 'L11' will result in hits only if those documents contain the specified descriptors and subdescriptors and if the descriptors share a common link and if the subdescriptors also share a common link. The third option for linking subdescriptors is 'L22'. When using this option, the links of each subdescriptor are 'added to' the links of the associated descriptor similar to option 'L12' above. Then the 'ACCUMULATIVE' linking is used. This means that for each hit the first descriptor/subdescriptors set of links must match the second set in at least one link, the third with the first or second set, and so on.

2. Sequence

It is possible to specify that descriptors appear in a record in a certain sequence. That is, one may require that descriptors A, B, C, and D appear in a record in that sequence. Several possibilities are apparent, as can be seen from the illustration below.

1.	---A---B---C---D---	descriptors in record, in sequence with intervening descriptors (non-contiguous)
2.	---ABCD------------	descriptors in record, in sequence, and are contiguous
3.	---D---C---B---A---	descriptors in reverse sequence, non-contiguous
4.	------DCBA---------	descriptors in reverse sequence, contiguous

This request language recognizes four ways of specifying sequential searches. They are:

	TYPE	RETRIEVES
1.	FC (forward contiguous)	Case 2 only
2.	FN (forward noncontiguous)	Case 1 and Case 2
3.	EC (either forward or reverse contiguous)	Case 2 and Case 4
4.	EN (either forward or reverse noncontiguous)	All Cases

A sequential search is specified by enclosing the desired descriptors in parentheses, preceded by one of the four codes shown above (FC, FN, EC, or EN) as below.

EN<A&B&C&D>

As with link searches, the AND operator must appear between descriptors, and no other operator is valid. No other parentheses may appear within those enclosing the sequential search descriptors.

Simple and Complex Descriptors

A simple descriptor is what the name implies, a descriptor alone, without qualification by either subdescriptors or numeric values.

A complex descriptor is one which is qualified, or modified in some way, by either subdescriptors or numeric values, or both. Some possible patterns of complex descriptors which may appear in a document are shown below:

(SD= subdescriptor and V= numeric value)

1. DESCR A; SDA; SDB; SDC;...SDN

2. DESCR A; V1; V2; V3; ...VN

3. DESCR A; SDA; V1; SDB; SDC; V2; SDX...

4. DESCR A; V1; V2; V3; V4; SDA; SDB; V5; V6; SDC; SDD; V7; V8; V9...

Subdescriptors always are associated with the preceding descriptor. Numeric values always apply to the immediately preceding subdescriptor, if any. If there is no subdescriptor, the value applies to the descriptor. Note that this requires that a numeric value which does pertain to a descriptor must follow immediately after that descriptor, and precede any subdescriptors which may also pertain to that same descriptor. In example four above, for example, values V1, V2, V3, and V4 apply to descriptor A, while values V5 and V6 apply to subdescriptor B, and values V7, V8, and V9 apply to subdescriptor D.

Structure of Descriptor/Subdescriptor Patterns

Inquiries involving descriptors and subdescriptors in combination are written as shown below.

DESCRIPTOR; SUBDESCRIPTOR; SUBDESCRIPTOR; SUBDESCRIPTOR ...

As many subdescriptors as desired may follow a descriptor. The descriptor is written first. Each subdescriptor must be preceded by a semi-colon. Thus, there is no semi-colon following the last subdescriptor. Spaces before, after, or on both sides of the semi-colon are permitted but not required.

All subdescriptors following a descriptor in an inquiry must be present in the document record if the document is to be a 'hit'. This, in effect, means that an 'AND' relationship is implied between a descriptor and its associated subdescriptor(s). No parentheses may be used in a subdescriptor string.

One may not phrase an inquiry such as 'DESCRIPTOR A with SUBDESCRIPTOR B or SUBDESCRIPTOR C' as a single complex expression. That is,

DESCR A; <SDB|SDC>

is not permissible. The question must be written as:

DESCR A; SDB | DESCR A; SDC

In phrasing subdescriptor questions, it will be convenient to treat the semi-colon preceding a subdescriptor as meaning 'with' (or 'and with'). Thus, for example, the question:

A; SDX; SDY & B; SDZ

Would read as:

'Descriptor A with subdescriptor X and with subdescriptor Y and descriptor B with subdescriptor Z'

Truncation

It is possible to make inquiries of the system using only partial descriptors by use of the truncation feature. In the normal, or untruncated, mode a descriptor in the inquiry must match exactly with a descriptor in the record order for the record to be a hit. Using the truncation feature, it is possible to ignore terminal characters of descriptors in the record. In the following inquiry, for example,

ROADS&ACCIDENTS

the descriptor 'ROADS' and the descriptor 'ACCIDENTS' must appear exactly as written in the document in order to be a hit. In,

ROAD$&ACCIDENT$

however, any descriptor such as ROAD, ROADS, ROADWAY, and the like, would match ROAD$. Similarly, ACCIDENT, ACCIDENTS, and ACCIDENTAL would match ACCIDENT$. Truncation is thus a useful feature for avoiding the necessity of 'ORing' together singular, plural, adjectival forms of the same word or words. The program will ignore any characters appearing in the document descriptors beyond those given in the inquiry. The truncation symbol (the $) acts to block further comparison.

Truncation should be used with care, however, as unexpected retrieval might result from its injudicious use. For instance, in the above example, ROAD$ will match successfully against not only the terms given above, but also against ROAD SIGNS, ROAD BUILDING, and any other descriptor, regardless of length which begins with the four characters ROAD. Similarly, ACCIDENT PREVENTION, ACCIDENTAL LOSS, and any other descriptor whose first eight characters are ACCIDENT will match successfully against ACCIDENT$.

The truncation feature may be used for both descriptors and subdescriptors. The only constraint is that there must be a minimum of one character preceding the $. An inquiry such as:

ROADS&$

is invalid and will be rejected by the system.

The exact position of the truncation symbol has an effect on the retrieval. ACCIDENT$ does not have the same meaning as ACCIDENT $. In the former case (no blank between the word and the $), the eight characters ACCIDENT are compared against the document records as described above. In the latter case, the nine characters A-C-C-I-D-E-N-T-blank are compared against the record. In this case, only such descriptors as ACCIDENT PREVENTION, and other multiple word descriptors beginning with the word ACCIDENT would be retrieved. Neither ACCIDENTS nor ACCIDENTAL would be retrieved. Furthermore, not even the simple descriptor ACCIDENT would be retrieved, because descriptors are assumed to end with the last significant character. That is, they do not have trailing blanks. Therefore, there would be no match between the eight character ACCIDENT in the record, and the nine character ACCIDENT $ in the inquiry.

Structure of Descriptor/Numeric Value Patterns

Descriptors qualified by numeric values are written using the operators shown below. Each operator precedes its numeric value. Note that the colon following each numeric operator is part of that operator and must be included:

1.	EQ:	Equal
2.	NE:	Not equal
3.	GE:	Greater than or equal
4.	GT:	Greater than
5.	LE:	Less than or equal
6.	LT:	Less than

A descriptor with a numeric value is written as below:

DESCR A; EQ: XXX

The numeric operator (EQ:) and the value itself (here represented as xxx) must be preceded by a semi colon. The numeric value may contain up to nine significant digits, and is written in the conventional form. If no decimal point is given, it is assumed to follow the last digit of the value.

The operators EQ: and NE: may never be used with any other operator. GE:, FT:, LE:, and LT: may be paired with one of the opposite type or may be used alone. All allowable combinations of numeric operators are shown on the next page.

1. EQ: XXX
2. NE: XXX
3. GE: XXX
4. GT: XXX
5. LE: XXX
6. LT: XXX
7. GE: XXX LT: YYY or LT: YYY GE: XXX   XXX must be less than YYY
8. GE: XXX LE: YYY or LE: YYY GE: XXX   XXX must be less then YYY
9. GT: XXX LT: YYY or LT: YYY GT: XXX   XXX must be less than YYY
10. GT: XXX LE: YYY or LE: YYY GT: XXX   XXX must be less than YYY

Note thatwhen paired numeric values are used, as in GE: XXX LT: YYY, there is no semi colon between the first numeric value and the operator preceding the second value. The lack of a semi colon here means that a single value, which falls within the desired range from GE: XXX TO LE: YYY is desired. A semi colon between GE: XXX; LT: YYY means that two values are desired, one of which is GE: XXX; the other of which is LT: YYY.

Numeric Value Precision

The precision of the numeric values used (i.e., the number of decimal points included) will affect the search results of using either the NE: or the EQ: operators.

When using GE:, GT:, LE:, or LT:, the numeric value given in the inquiry is extended to the system maximum of nine significant digits by adding trailing zeros as required. 2.15 becomes 2.15000000. Values in the document record are compared against this extended number. However, for EQ: and NE: the test for equality/inequality extends only to the level of precision given in the inquiry. That is, in

DESCR A; EQ: 2.15

any value from 2.15000000 through 2.15999999 following descriptor A is a hit.

Similarly, in

DESCR A; NE: 2.15

any value from 2.15000000 through 2.15999999 is a no hit.

Structure of Descriptor/Subdescriptor/Numeric Value Patterns

Both subdescriptors and numeric values may be used with descriptors as long as the appropriate rules are observed.

DESCR A; EQ: XXX; SDA; LE: YYY; SDB; SDC; GT: ZZZ LE: QQQ

is a valid statement for example.

SEARCHING SYSTEM

Purpose

This program is designed to accept requests from users, to validate those requests against the dictionary, to search the data files of the system, and to produce copies of those records in the file which meet the parameters of the search request.

Input

The input consists of request cards (outlined in search language description) and the files to be searched.

1. Tape

   The identification file tape is mounted on SYS007.

2. Disk

   a. The cross reference disk file is mounted on SYS018.

   b. The inverted file disk is mounted on SYS014.

   c. The dictionary disk file is mounted on SYS010.

3. Data Cell

   The master file data cell is assigned to SYS031.

Processing

This program operates in three logical phases. The first phase is the editing of the input request, to ensure that all parameters are satisfactory. Errors found in this validation operation will result in cancellation of the search attempt. For example, the use of the term in the request which is not an authorized descriptor, as determined by a search of the dictionary, would render further processing futile, since there is no such term in the file. Therefore, while the program will continue to edit the remainder of the terms in the request (to detect additional errors, if any), no searching will be performed. The processing of that erroneous request will be terminated and appropriate messages printed.

If the request is valid, however, the program proceeds to the actual searching. The usual form of search is the combined file method, making use of the inverted file on disk and the master file on the large capacity direct access device.

The second phase of the program, in this usual mode of operation, consists of a search of the inverted file which is an index to the master file. By examining the strings of document numbers following the descriptors in the request, the program is able to produce a 'candidate list' of potentially responsive documents. This list represents the output of this second phase.

That list is passed on to the third phase, the searching of the master file itself. Each document in that list is retrieved from the storage device and is examined in detail to determine whether or not it does, in fact, satisfy all parameters of the request. If it does, it is printed, written on tape, or both, as directed by the requestor. If it does not, it is ignored and the program examines the next member of the list. This process continues until the list is exhausted.

As described in 'search language description', there are several variations of this searching method. One such variation is an inverted file search only. In this method, the search is terminated at the end of the inverted file, producing only the candidate list in the form of a list of document numbers.

Another variation is the ID. file search. The ID. file is a subset of the master file containing only certain selected descriptors and no free text. If a search involving very common terms is desired, it may be preferable to search this ID. file since in certain circumstances it will be faster than a search of the master file.

It is also possible to search the tape, rather than the direct access version of the master file itself. This form of search is available for two reasons. First, it offers a back-up method of searching if the direct access device becomes unavailable for any reason. Second, FDA may wish to search its files at some computer installation which does not have the necessary storage devices.

Minor variations of the basic search technique included the 'document number search' which is simply a request to extract a given document or group of documents based upon their document numbers, and the 'mixed search' mode which is a combination of the document number search with the customary logical search. Both are explained in more detail in the 'search language description'.

Output

Output is of two general kinds. If a request contains errors which make a successful search impossible or highly improbable, messages to this effect are printed, and the search for that request terminated. If there are no errors, the output consists of either or both of the following:

1. A printed listing of documents which satisfy the parameters of the request. A variable amount of information may be printed, depending upon the requestor's specifications.

2. A tape record of some or all of the searchable section of documents which satisfy the request parameters. It is suitable for further processing via report programs or other special purpose programs which may be written. This tape must be mounted on SYS003.

   A scratch disk must be mounted on SYS020. It is used for inverted file document number strings.